April 16, 2020

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSTITÄT MÜNCHEN

App-Entwicklung und Marketing-Automatisierung

App Development and Marketing Automation

Enhancing Marketing Automation Processes with Push Notifications

ABSCHLUSSARBEIT

des Aufbaustudiums Informatik

von
JULIA PÜHL

Betreuerin der Universität:	Prof. Dr. Anne Brüggemann-Klein
Betreuerin der Firma:	Katharina Meitzler
Firma:	ec4u expert consulting ag
Abgabedatum:	16.04.2020
Matrikelnummer:	03699893

I confirm that this thesis is my own work and I have documented all sources and material used.

Munich, 16.04.2020 Julia Pühl

Abstract

It is important for companies to have a good relationship with their customers. Recent strategies in digital marketing emphasize that for optimizing this relationship, a company has to focus on the customer's perspective. This way, a company can figure out when a customer wants to be approached how, which is what is described by the term Customer Relationship Management, short CRM. In today's digital and interconnected world, there are more ways of interactions than ever, which offers great opportunities to CRM, but also challenges. This thesis investigates a form of communication between customer and company, that is not that common yet in digital marketing, which are push notifications.

The goal is to identify possible use cases for push notifications in digital marketing. The use cases are first determined theoretically by collecting different types of push notifications and how they can be integrated into CRM strategies. It is also discussed what benefits can be achieved by integrating them. In the practical part, an iOS app is developed with the primary purpose to test the use cases of push notifications. For the testing, a set of use cases is selected in the context of event management. The app is integrated into existing CRM software, the Salesforce Marketing Cloud, which generates the push notifications and provides customer data. With this setup, push notifications for the specified use cases are created and evaluated.

Contents

1 Introduction

Digitalisation affects a broad range of fields and is a much-used term. Chapco-Wade (2018) defines Digitalisation as "the use of digital technologies [...] to [...] transform how customers and companies engage and interact". According to this definition, Digitalisation changes the communication between customers and companies. This communication is a central part in marketing. Consequently, Digitalisation plays a significant role in marketing, which is why the term 'digital marketing' got established. Throughout this thesis, one specific form of digital marketing will be investigated, the communication through push notifications via an app.

1.1 Motivation

Online advertisement, messaging through social media or emails... digital marketing is already being performed via various channels. Why would another form of digital interaction be necessary? The motivation behind push notification via an app as additional communication channel is founded in three factors. Primarily, there is the issue, that existing channels partly fail to reach customers, which might result in a loss of customers. Push notifications offer options, that other communication channels do not have, and with those, customers could be prevented from churning. More positively seen, these options could help reaching more or other customers at better moments via new forms of interactions. Finally, Digitalisation itself is a driving force, which, through its dynamic nature, requires companies to constantly adopt and try out new technologies in order to stay competitive and meet the customers' expectations.

1.2 Goals

The goal of the thesis is to investigate scenarios in which push notifications can be used and in what way they can provide an added value considering existing forms of digital marketing. Accordingly, the **research question** is the following:

Which use cases (with an added value) for push notifications via an app as communication channel in digital marketing can be identified?

1.3 Outline

This research question will be approached in a theoretical and a practical part. In the theoretical part, the potential added value of push notifications will be analysed and their use cases will be identified. The practical part includes the development of an app, which can then be used for testing to verify the identified use cases.

2 Background

2.1 Digital Marketing

In this chapter, common concepts in digital marketing are introduced, which push notifications and other communication channels are based on.

2.1.1 Marketing and Customer Relationship Management

The context of digital marketing, in which push notifications will be placed, will be considered and explained in this chapter. Starting from a very broad view, the field that all processes that will be discussed belong to, is marketing. According to Peter Ducker, a famous American management scholar, marketing is so fundamental that it encompasses the entire business and therefore cannot be narrowed down to a specific activity (Li and Ma, 2015). Others define marketing as any activity performed by a company to get the customer to purchase the product (Tardi, 2019). Wikipedia provides a more specific description of marketing stressing that it is about creating relationships with the customer with the goal to satisfy them (Wikipedia, 2019). One might argue about which activities are part of marketing, but an indisputable characteristic is that they are all based on interactions between company and customer. Therefore, relationships to the customer play a crucial role in marketing. The concept dealing with and improving these relationships is called Customer Relationship Management (CRM), which will be explained in more detail in the following.

CRM processes are meant to optimize the relationship to the customer for a higher customer satisfaction. This is achieved by acknowledging the customer and their needs as important resource and decisive force and by centering all processes around them (Li and Ma, 2015). Why is this profitable for a company? A higher satisfaction among customers means a higher revenue for the company. Li and Ma (2015) list interesting facts supporting the importance of customer satisfaction for a company: one satisfied customer brings eight potential businesses whereas one dissatisfied customer can dissuade 25 customers from a purchase. Furthermore, they state that a small increase in customer satisfaction of five percent will cause a boost in the company's profits by a hundred percent. Another considerable observation is that a very satisfied customer is six times more willing to buy than a satisfied customer. By creating a high customer satisfaction, CRM strategies extracts value from the customer that helps the company to increase their profit (Lemon and Verhoef, 2016). From this statement, it becomes clear that ultimately, CRM focuses on creating value for the company. However, if customer satisfaction is crucial for a company's success, what is it, that creates value for the customer? Creating value for both, the customer and the company, might seem conflicting at first, but understanding that one causes the other reveals that it can be achieved through the same strategy. As described by Lemon and Verhoef (2016), companies shifted their perspective only recently, moving their focus from how customers can create value for the company to how the company can create value for the customer. An answer to the question, how value for the customer is created is given by Abbott (1955), who stresses that people do not desire products, but satisfying experiences. With that in mind, many firms set creating a strong Customer Experience as their main management goal (Lemon and Verhoef, 2016). In the following, the concepts behind the Customer Experience will be presented.

2.1.2 Customer Experience and Customer Lifecycle

Since an experience can be quite abstract and subjective, the term Customer Experience will first be defined. Generally speaking, it encompasses all of the customer's responses to every contact with the company (Lemon and Verhoef, 2016). These responses can be of different kinds yielding five types of experiences: responses in form of sensory stimuli, feelings (affective), thoughts (cognitive), actions (physical) or social identification. Not only encompasses the Customer Experience all kinds of responses but also the whole range of a company's offering, reaching from advertisement and packaging of the product over the product itself to its ease of use and reliability. The holistic nature of Customer Experiences is important to keep in mind and determines the way high Customer Experience can be achieved: on the one hand, it requires every contact point with the company, also called touch point, to be identified and managed. On the other hand, customers in all kinds of different phases have to be considered. These phases correspond to the aspect of the company's offering, a customer is engaged with, for example the advertisement of a product. Typically, every customer runs through a set of phases throughout their engagement with a company, also called the Customer Lifecycle. It starts with the customer realizing their need of something, triggered by an issue or emotion, continues with the customer buying the product and finally ends with them using it ("The Essential Guide to the Customer Journey and Lifecycle | Customer Success Software | Gainsight", 2020). This process corresponds to the three main stages Pre-Purchase, Purchase and Post-Purchase. Each of these stages include multiple steps as depicted in the following table:

Table 1: The Customer Lifecycle

main phase	**sub-phase**	**details**	**behaviors**	**focus on**	**additional info**
Pre-Purchase "I need"	awareness	need generation	recognizing a problem and searching for a solution or following a suggestion/ an emotion		
	consideration	overview over products, product selection	gathering information and analyzing/ evaluating it		
Purchase "I buy"	purchase	comparison of prices, purchase, product delivery	choosing, ordering and paying	shopping experience, avoiding choice overload, purchase confidence, decision satisfaction, completion of purchase	temporally compressed, but crucial
Post-Purchase "I use"	use	commissioning and use	using/ consuming/ engaging with the product, requesting the service	consumption experience, service recovery, return products, repurchase, word of mouth	product becomes touchpoint!
	loyalty	use with opinion		loyalty loop: trigger leading to loyalty or restart of the lifecycle	

Knowing the customers current lifecycle phase and knowing his or her opinion enables the company to adjust their actions to the customer's phase and this way a better Customer Experience can be created (Lemon and Verhoef, 2016). Tthe motivation behind this method will be shortly explained. Companies approaching their customers in a too direct way from the limited perspective of the company has been identified as a main reason for failed interactions (Litmos video). If the actions of a company do not correspond to the situation of the customer, the customer is provided with wrong information at the wrong time leading to them feeling overwhelmed or annoyed. Hereby, not only the content and time of the interaction play an important role, but also the channel through which it happens. Recently, the variety of communication channels has increased significantly because of the integration of digital media. This holds many possibilities, but also complicates the estimation of the current lifecycle phase of the customer, creates large volumes of data and leads to the company losing parts of its control. Because of these developments, methods are required that help managing the Customer Experience as will be discussed in the next chapter.

2.1.3 Marketing Automation

Marketing has been defined in the beginning of this chapter as fundamental factor in every business and as element in the value chain. Its activities are based on interactions between company and customers with the goal to get customers to purchase the product by meeting their needs and expectations. For successful marketing, a company has to consider the marketing mix made up of the four Ps of marketing, which are Product, Price, Promotion and Place (Rosenbloom, 2013). In this chapter, the focus will be put on marketing automation.

The term marketing automation implies that marketing processes previously controlled manually become automated. However, marketing automation involves more than that: next to automating marketing activities, it can track leads across all channels and proves the value of marketing efforts by providing closed-loop reports on the effectiveness of the applied methods (Sweezey, 2014). This already suggests that the scope of marketing automation is rather broad. It can be split into three main parts: its functionality on the customer side, that captures and makes sense of customer data, its functionality on the company side, that uses this data to create personal campaigns at scale automatically and finally the reporting functionality that makes a connection between both sides by associating the data with the achieved revenue. The crucial characteristic of marketing automation, that makes all this possible, is its consistency across all channels and processes (Salesforce). There are no longer separate tools and datasets, but one single platform for all marketing activities.

Having defined marketing automation, its role within the already existing structure of marketing will be discussed in the following. Marketing automation is strongly related to online marketing. Sweezey (2014) calls it an extension of online marketing and states that they are in a symbiotic correlation with one another. Marketing automation needs online marketing to work and online marketing activities are made more effective by marketing automation. Before going into detail about online marketing, it will be explained why among multiple types of marketing, the focus is put on online marketing. Online marketing dominates marketing due to the fact that the average consumer spends 12 hours in front of a screen daily (Fisher, 2019). As a consequence, in 2019, 61 percent of purchases by people at the age between 22 and 26 were made online (Mezzacca et al., 2019). 246.5 billion emails were sent per day worldwide in 2019 (The Radicati Group, Inc., 2015). These statistics stress how important it is for a company to be present online and therefore to develop various online marketing strategies. Examples for these strategies are Search Engine Optimization (SEO) and Search Engine Marketing (SEM), which are both part of search marketing. Other subcategories of online marketing are content marketing, social media marketing, mobile marketing and email marketing.

As already stated, marketing automation makes online marketing more effective. The following will explain how, based on concrete activities of the different online marketing subcategories. Starting with search marketing, Search Engine Optimization (SEO) and Search Engine Marketing (SEM) both capitalize on the consumer's searches. SEO is doing so by tracking leads and refers to the natural way a company ranks in searches, while SEM tracks keywords in the context of the paid listing in searches. Google, for example, made a revenue of 116.32 billion dollars from its ads in 2018, which leaves no doubt that this strategy can bring a lot of success ("Google: ad revenue 2001-2018 | Statista", 2020). However, which activities contributed to this success and which ones were obstructive is not always clear. Marketing Automation

can answer this question via its closed-loop reporting, that proves the Return on Investment (ROI) of certain search marketing activities.

For content marketing, marketing automation can provide support by automating the distribution of content and by facilitating the follow-up of leads who got engaged with the content. This way, more time can be spent on the generation of content (Sweezey, 2014). An optimized generation and distribution of content is important to improve marketing because to a company the content is their path to engaging with the customer, who, in return, is looking for helpful and relevant content.

With social media marketing, the challenge is to manage the massive level of communication. The great characteristic of social media is its huge reach. However, with this opportunity also comes the risk of losing the overview. Marketing automation helps to solve this issue by keeping a consistent message across different channels and by supporting the communication management through automated programs. Besides that, it can tie social activities to revenue, which proves the value of social media (Sweezey, 2014).

Mobile marketing is, strictly speaking, not a separate marketing type, since it only differs in the kind of devices consumers use to interact with the content. It is worth considering it separately though because of its growing extent: according to the CouponFollow Millennial Shopping Report 2019, 16 percent of the purchases by millennials were made on mobile devices in 2017, which increased to 36 percent in 2019 (Mezzacca et al., 2019). Social media is nearly exclusively used on phones. Also the fact that more people have phones than have electricity or access to drinking water illustrates, from a marketing perspective, the huge reach provided by mobile devices (Sweezey, 2014). Mobile marketing can be implemented in multiple ways, via an app or, alternatively via mobile versions of existing online content, for instance a website or the inbox. Either way, the key of mobile marketing is to provide relevant content at the correct time. Mobile channels make this possible supported by marketing automation. Marketing Automation can identify relevant content and the right moment via its tracking, which holds information about keywords, leads and interactions. Specifically for emails, this means that instead of manually sending blast emails that are disconnected from the customer's interactions, with the help of marketing automation, emails are personalized campaigns sent automatically and at scale and can dynamically react to change.

For creating relevant content at the right time, the customer's phase within the lifecycle should be known. This information can be derived from the data tracked via marketing automation. Without marketing automation, it is difficult to determine the customer's phase because data is missing or has no connection to a context. Previously, meaningful data about the customer would only be available after they made a purchase, which was mainly helpful for the sales department. Via marketing automation, it is possible to 'know' the customer from the very start of their lifecycle, long before they make a purchase. A general information about the average consumer in the starting phase of the lifecycle is that they are hypereducated and hypersensitive (Sweezey, 2014). This is attributed to fact that there is a massive amount of information available online that, on the one hand, consumers use to educate themselves before purchasing a product, and on the other hand, leads to the customer only considering content that is a hundred percent relevant to them. Another fact is, that 93 percent of consumers enter the lifecycle via a search online (Sweezey, 2014). With this information, campaigns and activities can be adjusted accordingly. How marketing can be optimized considering the different kinds of communication channels will be discussed in the following chapter.

2.1.4 Communication Channels in Marketing

As laid out in the previous chapter, to be able to optimize marketing, the customer should be characterized as well as possible and their phase in the lifecycle should be known. For this, the full Customer Lifecycle has to be considered. More concretely, this means that every touch point along the lifecycle has to be identified and designed in accordance to the customer. At every touch point, the customer interacts with the company through a certain channel. Extracting customer data from the channel helps the company to understand the customer and thereby enables it to create a better Customer Experience (Lemon and Verhoef, 2016). Selecting the appropriate channel to communicate with the customer is crucial in marketing, which is why communication channels will be discussed in this chapter.

Before going into detail about marketing channels, touch points will be discussed. They describe every moment throughout the customer lifecycle, in which the company gets 'in touch' with the customer or vice versa. The channel is the medium that makes the contact possible. There are touch points designed and controlled by the company, the brand-owned touch points, and there are touch points, which are customer actions and cannot be influenced by the company or others, the customer-owned touch points (Lemon and Verhoef, 2016). Examples for the latter are the customer's thinking of their own needs in the pre-purchase or their choice of a payment method in the purchase phase. Brand-owned touchpoints can be promotional media, like ads, websites and loyalty programs, or elements of the marketing mix, such as the packaging.

A marketing channel can be used at brand-owned touchpoints, however its definition implies that it is not solely controlled by the company. From the perspective of marketing management, the marketing channel is defined as "external contactual organisation that management operates to achieve its distribution goals" (Rosenbloom, 2013, p.28). Contactual organisation means parties involved in negotiatory functions, like buying or selling. This implies that a marketing channel can include multiple members: the manufacturer, advertising agent, wholesaler, retailer and the consumer. The way they work together make up different flows with different tasks and goals. Among them, there is the promotion flow, that describes the persuasive communication that happens between manufacturer and consumer and all parties in between including the advertising agency, wholesalers and retailers.

Recent developments in Internet and Communication Technologies (ICT) have led to an altered structure of marketing channels due to the integration of digital media and e-commerce (Rosenbloom, 2013). New channels have emerged, like the online and mobile channel and social networking. Push notifications via an app can be classified as mobile channel, which is why this channel will be investigated more thoroughly. The mobile channel is most suitable for the pre-purchase phase because it facilitates searching for the product and gathering information about it (Lemon and Verhoef, 2016). However, it is also used in other phases and, due to its flexibility, often in combination with other channels. An example for this is "showrooming", which describes the phenomenon of consumers making a purchase online after or even while looking at the product in a real store (Rapp et al, 2015). Next to enhancing cross-channel synergies, mobile channels also offer many new possibilities: offers or information can be made time- or location-sensitive and this way more personal. A difference in using mobile channels, compared to other channels, is interacting with a device via a touch screen. Being able to touch an item or button on screen influences the customer behavior: touching the product creates a greater sense of ownership and attachment than clicking it. Furthermore, it has been found that direct touch interfaces make consumers search for more alternatives. Another interesting finding is that touching a button as a way of expressing preferences, compared to verbalizing them, influences the consumer's self control (Brasel and Gips, 2014).

The integration of mobile and other new channels that emerged from developments in ICT, affects both, the company and their customers. With increasing opportunities also the customers' expectations increase (Rosenbloom, 2013). They anticipate that products and services are accessible from wherever, whenever and however they want through more and better channels. They assume to always be able to make the optimal channel choice. This brings up the question, what the customer's channel choice is influenced by. The customer's decision on a channel depends on his or her preferences and on the properties of the channel and how they match with the customer's phase. For example, a customer in the pre-purchase phase might use a mobile channel, while for buying the product he or she might prefer using a desktop. This will be discussed in more detail in the following chapter with respect to push notifications. Apart from this, channel choices also influence each other through mechanisms like channel inertia, lock-in effects and cross-channel synergies (Konus et al, 2008). Among the heightened expectations of the customer, there is also the claim to be able to use multiple channels either throughout the different phases or simultaneously. A customer who searches in a different channel than where he or she makes the purchase is called a "Research Shopper". Research Shoppers can be showrooming meaning they search in store and buy online, as already explained earlier, or vice versa, they search online and buy in store, which is called webrooming.

The company, on the other side, has to be able to meet and even exceed these customer expectations. The key to achieving this is the right multi-channel strategy. Hereby, it is important to keep in mind that, for an effective mix of channels, not quantity, but quality is important. As important is to understand how these channels interact with each other to be able to identify synergies and conflicts. Per definition, multi-channel synergies occur if one channel enhances the efficiency and effectiveness of other channels (Rosenbloom, 2013). This happens, for example, in case of "webrooming", where the online channel supports the conventional one by providing additional information. Another example is bringing in other channels where a stocked-out product is still available. However, using multiple channels also holds the risk of creating a conflict. Such a conflict can emerge if different channel members both make use of the same kind of channel. Concretely, if a manufacturer and a distributor both sell via an online channel, the manufacturer's online channel takes away customers from the distributor's online channel and vice versa resulting in a zero-sum game (Rosenbloom, 2013). Conflicts like these can be avoided by knowing and managing the factors influencing the channels across all channel members.

Regardless of the risk of creating conflicts, multi-channel strategies are indispensable for companies, not only to meet customer expectations but also because they are means for gaining a sustainable competitive advantage. A company has a sustainable competitive advantage if a property of its offering cannot easily be copied by a competitor. In today's market, creating such a property has become extremely difficult because no matter which of the four P's of marketing, product, price, promotion and place, is focused on, differences can always be imitated or even exceeded. Especially in promotion it is challenging to stand out, considering the flood of messages and information consumers are exposed to daily minimizing the impact of each message. However, with the help of a well-formulated and complex multi-channel strategy, not only can consumers be reached more efficiently, but an advantage over competitors is gained because this strategy cannot easily be copied by them (Rosenbloom, 2013). Not only multi-channel strategies are a way of tackling increased customer expectations and the challenge to stand out in the overload of information. Another method is to focus on the personalisation of the provided content and moment of interaction. Using push notifications, this method can be realized particularly well because they are considered to be personal and can be individually adapted to the consumers in various ways. Push notifications will be presented in detail in the following chapter.

2.2 Push Notifications

2.2.1 Types, Triggers and Content of Push Notifications

Push notifications exist as app and as web push notifications. Web push notifications are pop-up messages shown in an active browser. They are sent via a website or web app to desktop or mobile devices, meaning that they are device-independent. Furthermore, users can receive web push notifications even when they are not on the respective website. However, web push notifications only work if the browser is active. Opposed to that, app push notifications can also be sent while the app is closed. They require, though, that the user installed the app and allowed it to send push notifications. Next to that, app push notifications are restricted to mobile devices ("What are Push Notifications? A Complete Guide [2020]", 2020). A comparison between the properties of these two types of push notifications is shows in the table 2.

Table 2: The Properties of Web and App Push Notifications

Web Push Notifications	**App Push Notifications**
on mobile and desktop devices	only on mobile devices (and dependent of their operating system)
sent to everyone who allowed push notifications	sent to everyone who installed the app and allowed push notifications
require browser to be active	work even if app is closed
only simple interactions possible (eg clicking)	more variate interactions possible (eg swiping)

Summarizing the differences between web and app push notifications, app push notifications come with more restrictions since they require the user to install the app and they are device and platform dependent. Therefore, web push notifications might have a broader reach. However, assuming, that the restrictions associated with app push notifications can be overcome, they offer more effective options because the user can receive them if he or she did not open the app and can interact with them more diversely.

One of the fundamental ideas behind push notifications is that they are induced by factors that are related to the user without them being actively aware of it. Factors that can trigger push notifications will be described in the following. They can be split into four main categories: updates, location, time and promotions.

Updates that trigger push notifications can be of different types. An update can refer to something that happened within the app, like a message or comment on a post or a like ("8 Types of Push Notifications To Reinforce Your Business", 2020). A very common example for this is WhatsApp showing push notifications as previews of received messages. An update can also refer to something that happened in real life. A fitness-app-user ran 10 kilometres and by that achieved their personal goal set in the app. The app then sends a push notification congratulating the user and encouraging them to keep going. Updates on events in real-life are not only activities performed by the user, but also occurrences that happen around them. Push notifications can inform the user about them in form of news or weather forecasts

(Dubey, 2019). More supportive than informative are push notifications reminding the user of upcoming events or responsibilities that need to be done. For example, a travel app can let the user know, that it is time check-in on their flight by sending a corresponding push notification (Zantal-Wiener, 2019). Another type of updates are updates on processes. A typical process, which a user wants to be updated on reliably is the payment process. A less serious, but relevant example is the order delivery process (Zantal-Wiener, 2019). Push notifications can include updates on the current location of the order or on the estimated delivery time. Next to that, the app itself is also a changing process, which is why push notifications can also let users know that a new update of the app is available. There are two more cases, that can also be categorized as update push notifications: the "inactive user" and the "distracted user". In the first case, the user has not been using the app for a while and a push notification is aimed at encouraging the user to re-engage with the app. The second case means the situation, in which the user leaves an activity within the app unfinished, like for example filling the online shopping cart without proceeding to the purchase ("8 Types of Push Notifications To Reinforce Your Business", 2020). Redirecting users to their abandoned shopping carts is a very common purpose of push notifications.

Push notifications can also be triggered by the user's location. Users within in a certain area get a notification providing information that is related to this area. This information can be related to where to find the next grocery store or a recommendation for nearby restaurants or hotels (Zantal-Wiener, 2019). The respective area can be determined via geo-fencing or with the help of beacons. In geo-fencing, virtual perimeters are set, defining the area. Push notifications can be sent as entry message, triggered when users are entering the are, or as exit message to everyone leaving the area. Alternatively, a geographical area can be defined through a beacon. Whenever a user has a certain proximity to the beacon, he or she receives a push notification.

Promotional offers are another kind of trigger of push notifications. Promotional offers can be discounts, sales or giveaways. Promotion-based push notifications are especially useful for offers, that are only valid within a certain time frame ("8 Types of Push Notifications To Reinforce Your Business", 2020). Next to that, they are an effective way of notifying users if a price is dropped below a certain value. This applies to services or products with fluctuating prices, like for example flights.

Another trigger of push notifications is time. The time for sending a push notification can be set in beforehand once or recurrently. Time frames can also be used to determine when push notifications should not be sent, which is what is known as "do-not-disturb-mode" (Zantal-Wiener, 2019). Time alone as trigger is not particularly special, but when combined with the user's personal data and other triggers, it can create a great impact. For example, knowing the user's name and date of birth, a push notification can be triggered on their birthday including personalized congratulations and a gift voucher.

In general, combining different kinds of triggers plus including personal data is what makes push notifications most popular (O'Connell, 2016). According to O'Connell, geo-based push notifications including related offers are third most preferred. For example, receives a special discount via a push notification for a nearby store. Adding information on recent events to location-based promotions can create even more effect. Such an event could be a storm that caused the train connection, the user booked, to be cancelled. Consequently, a push notification could ask the user if he or she wishes to extend their hotel stay by one night for a reduced price (Dubey, 2019). O'Connell states that the most preferred push notifications are the ones providing special offers according to the user's preferences. These preferences were either entered by the user directly, for example as maximum price or favorite brand, or collected as data from the user's interaction with the app.

The personalisation of push notifications plays a crucial role for their effectiveness. Especially for promotional push notifications, it is important to not let them sound too "salesy", since this may cause the user to be put off by them (Dubey, 2019). Therefore, rather than asking the user to do something, e.g. to check out an offer, a better approach is to present this offer as a favor to them or to provide it as an opportunity. To be able to do that, personal data is necessary that is integrated into the offer and makes it match the user's situation, like an extra night at a hotel because a train got cancelled, as described earlier. These kind of push notifications make the user feel acknowledged and thought of, instead of being bothered and controlled (Zantal-Wiener, 2019). If push notifications evoke "feel-good-moments" as they are caused by interacting with a friend, the user will associate these positive emotions with the app and the underlying company.

2.2.2 Compatibility of Push Notifications with Marketing Concepts

In this chapter it will be evaluated, how push notifications are compatible with the different phases of the customer lifecycle. Cases in which push notification can be applied will be discribed. To recall, the customer lifecycle can roughly be split into three phases: the pre-purchase phase, in which the potential customer is in need of something and therefore is looking to buy a solution, the purchase phase and the post-purchase phase, in which the customer uses the purchased product.

Push notifications are a way of communicating via a mobile channel, or more concretely, via an app. Whether this type of communication is effective in a certain phase depends on multiple factors, such as the age of the target group, the country they live in or the type of industry the company offering the product is in. For example, people are more likely to buy fashion related products online rather than medicine (Statista Research Department, 2020). Accordingly, consumers in certain industry sectors are better reachable via a mobile channel than others. Obviously, the characteristics of the consumers play a role as well, as the younger generation is more experienced in using the internet on mobile devices and people living in countries with a higher internet penetration are more actively using their phones than others. Putting these deviations aside, the feasibility of push notifications will be evaluated for each phase of the customer lifecycle in the following.

In the pre-purchase phase, the consumers inform themselves and collect information on specific products. The mobile channel is often used for this, however, people mostly do their research in a mobile browser rather than in an app (Simoes, 2013). A possible explanation for this is that in this phase, the consumer wants to consider and compare different brands, which is not possible when using an app of one specific company. This makes applying push notifications in the pre-purchase phase challenging: it requires the consumer to have downloaded the app, which is not likely at this point because the consumer might not have decided on a certain brand and does not want to install their app. Assuming however, that the consumer is confident enough with the company to get their app, push notification can be very powerful in the pre-purchase phase. If a customer is debating whether to buy one product or the other, a push notification can literally give them the final 'push' that causes them the make a decision. Push notifications can provide the consumer with exactly the information they have been looking for about a product or they can offer them a promotional code triggered by a time or location. For example, a consumer can be close to a store of the company whose app they have on their phone and therefore gets a push notification including an offer on a product they have favored in the app. This push notification is

useful to the consumer because they do not have to invest time in searching for the best deal or for the company's store.

The question, whether push notifications can be effectively used in the purchase phase can be answered in different ways. On the one hand, statistics show that consumers prefer making the purchase on a desktop rather than on a mobile device (Carter, 2019). This indicates that push notifications are not very suitable for trying to get the consumer to purchase something. On the other hand, there is also research that found that the touch screens of mobile devices influences the consumer's decision-making process (Lemon and Verhoef, 2016). Consumers tend to be less hesitant to executing the purchase, if they are only one click away from owning the product. Furthermore, seeing the product on a touch screen and being able to click it and maybe also zoom in on it give the consumer the impression they already own it, which also speed up their process of deciding whether they want to buy it or not. Thus, there are cases in which push notifications can be very useful. For example, the consumer added an item to the cart in the app and then left without finishing the purchase. A push notification reminding them of their item in the cart, can bring the consumer back into the app to complete their order.

According to O'Connell (2015), one of the most effective use case of push notifications is to encourage in-active customers to re-engage with the company and their app. This is part of the post-purchase phase, which is about maintaining a relationship to the customer to build up trust and to eventually animate the customer to re-enter the lifecycle. Staying in contact with the customer can be achieved by engaging with them in different ways, for example by updating them on events, motivating them to keep up the good progress or by reminding them of an appointment. Using push notifications in these ways can demonstrate the benefit to the customer, they get from using the app, which increases the app usage, ideally also the in-app purchases and eventually strengthens the bond between customer and company (Khandelwal, 2020). The following table summarizes the use cases of push notifications throughout the customer lifecycle including the reasons why push notifications should or should not be used in a particular phase.

Table 3: Use Cases of Push Notifications throughout the customer lifecycle

	Pre-Purchase	**Purchase**	**Post-Purchase**
pro	mobile channel is used for browsing	touchscreens tempt the consumer more to make a purchase because being able to touch the item on screen suggest a feeling of ownership and clicking is quick and easy	activities in this phase match the purpose of push notification: to personally engage with the customer; relationship between customer and brand already exists -> customer is likely to be willing to get the app
contra	mobile browser is used more frequently than app because consumer's goal in this phase is to get an overview of various brand - app limits consumer to one specific brand and consumer might not even be willing to install the app	purchases are made on desktops more frequently than on mobile devices	
use case	promotional push notification corresponding to the current season/ to the consumer's location/ to certain dates, eg the consumer's birthday; informative push notification according to consumer's search activities/ preferences	reminding push notification after customer abandoned cart; limited-time offers	supportive push notification reminding customer of upcoming event/ appointment; encouraging push notification congratulating customer on their progress; entertaining push notification including news or updates relevant to the customer

From the table, it can be suggested, that push notifications are best suited in the post-purchase phase. One of the reasons for this is, that in practice, customers can only be reached via push notifications, if they installed the app. This is more likely for customers to do, who have already made a purchase and established trust with the company. At this point, it is important for the company to maintain the relationship to the customer. Push notification are very suitable for this purpose because they provide a personal way to engage with the customer. Applying push notifications for personal re-engagement also decreases the risk of mis-using them as 'promotional spam'.

2.2.3 The Potential of Push Notifications

In the previous chapter, it was analyzed *how* push notifications can be applied throughout the phases of the customer lifecycle. In this chapter, it will be discussed *why* push notification should be used. Their potential will be explained by pointing out their benefits, also with respect to the email channel, which is with 3.9 billion active email users worldwide (The Radicati Group, 2019) still among the most used marketing channels. Nevertheless, the effectiveness of emails is questionable since they have often been miss-used by sending them in large numbers including a default content. Not only the amount have emails are overwhelming, but in general the content that is available online. In order to not get lost in the flood of information, consumers have started to become hypersensitive, meaning they only give their attention to content that is one hundred percent relevant to them (Sweezey, 2014). Therefore, they should be approached using highly personalized messages. This can be realized particularly well with push notifications since they offer many options for adapting to the user. Not only can their content be

designed according to the user, but also the time they are sent can be individually aligned with the user's actions. An example is a push notification being triggered by the user's location as described in chapter 2.2.1. With this functionality, push notifications have an advantage over emails.

Another benefit of push notifications comes from the fact that they are mostly used by messaging apps. A friend texting is relevant and interesting to the user and subsequently, push notifications are associated with excitement. The sound of an arriving notification creates the urge to immediately check it. Push notification spark a direct reaction, which is special considering the huge amount of inputs from all kinds of sources that constantly compete for the user's attention. The fact that humans have a short attention span than a goldfish (Watson, 2015) expresses how easily distracted we are. Push notifications stand out from the mass of inputs, which is why they reach the user. This reachability via push notifications can also be addressed to the fact that the average user checks their phone every 12 minutes. This makes the phone's lockscreen the perfect medium for speaking to the user. While emails can be displayed on the lockscreen as well, they are still in many cases only paid attention to during working hours. Push notifications might not apply to these restrictions since they are perceived to be part of the personal context. Furthermore, push notifications can present a message in a more compressed and catchy way, which makes them particularly useful for delivering time-critical content, like short-term notices. A short and simple message additionally stimulate interest and a quick reaction in the user.

Users thinking of push notifications as something interesting and personal give them a big bonus over other channels, however this image can be damaged, if push notifications are not applied with care. The risk is, that push notifications are over-used and not enough effort is put into filling them with highly relevant content. Therefore, the use of push notification should be reserved for adequate cases, otherwise they will suffer, just like emails, form a lack of attention because their special effect is lost.

3 Existing Concepts

Marketing Automation is performed by using dedicated software. There are different vendors offering various levels of marketing automation technologies and one of them is Salesforce. Salesforce provides tools for Customer Relationship Management, Customer Analysis and Business Intelligence. These tools are integrated into one platform together with collected data. The tools and data cover all channels and departments, including marketing, sales, commerce and service, with the goal to provide a unified view on the customer and, from the customer's perspective, a seamless customer experience. For each department, Salesforce offers a certain set of tools and an associated cloud. As already described earlier, it as important to provide personalized, interactive content, provided at the right time to the right customer, through the right channel. In order to achieve this, access to coherent data has to be given, which is ensured by the cloud. From this data, meaningful information for the campaign's properties can be extracted, analysed and used for predictions with the help of automation processes. The artificial intelligence tool provided by Salesforce, that performs these automation processes, is called Einstein. For this project, the Marketing Cloud will be used because it includes a product called MobilePush which is part of the Mobile Studio and enables the implementation of push notifications. Next to this, there are more tools included in the Marketing Cloud, of which the most relevant ones are summarized in table 4.

Table 4: Salesforce Marketing Cloud Products

product	**description**	**goal**
Journey Builder	used to create personalized journeys including information about what content/moment is right for which customer; adapts to predictions and changes in behavior	consistent messages across all channels and departments
Email Studio	used to create personalized email campaigns containing responsive and relevant content, sent to particular customers that have been identified via data segmentation; interacts with customer throughout the whole journey	automated sending of emails at scale while keeping them individual and flexible
Mobile Studio	used to create personalized Push Notifications (MobilePush), Sms (MobileConnect) and group messages (GroupConnect) sent at the right time to the right customer including customized discounts/ offers (possibly enhanced with rich media); triggered by location/ event/ behavior/ preferences/ demographic data	enhanced customization via triggers determining the right time; increased In-App-purchase/ App-usage

4 Requirement Analysis

4.1 Overview

In the requirement analysis starts to specify the system that will be developed, which is an app receiving push notifications. First the systems in which the app will be integrated are presented. Then the app will be defined by determining its requirements. Based on the requirements, different scenarios and use cases will be developed for the app. Finally, the app will be represented by various types of models.

4.2 Current System

The will be connected to two already existing system, the Apple Push Notification server and the Salesforce Marketing Cloud.

4.2.1 Push Notification Service by Apple

One of the already existing systems, that will be used by the app, is the Apple Push Notification service (APNs). It will be integrated into the app to enable push notification. The following describes the APNs and how it works.

Four components are crucial in the process of sending and receiving a push notification with Apple: the device, in this case an iPhone, the app running on the iPhone, the provider server and the Apple Push Notification service (APNs). Latter is managed by Apple and supports the transmission of push notifications between app and server. The server, where push notifications originate from, is the Salesforce Marketing Cloud.

Before a push notification can be sent and received, the app first has to be registered with the APNs. This process is necessary for the server to be able to identify the device it is supposed to send the notification to. It happens every time the app launches and consists of the following steps: the app requests a token from the APNs, that sends one back to the app. After a network connection between app and server is established, the app can forward the token, together with other identifying info, like an email address, to the server. In case data of the same identity already exists in the database, it can be coupled to the token and data from the app. This way, the server can associate the token, representing the app on a device, with an identity stored in the database.

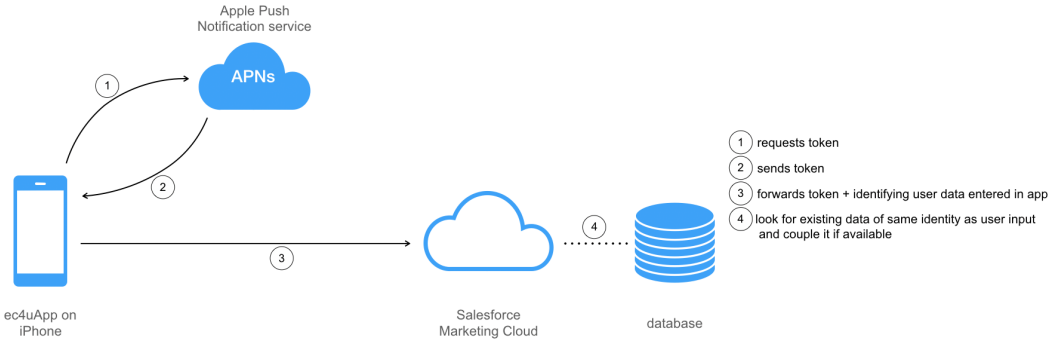


Figure 1: Registering an app with the APNs

Given these prerequisites, the server can now generate and send a push notification to the app over the APNs. For this process, the server first has to establish a secure connection to the APNs: the server

requests a connection to the APNs using transport layer security (TLS). Subsequently, the APNs sends a certificate, which previously was created with an Apple Developer Account, to the server. After the server has validated it, it sends it back to the APNs and the secure connection is established. If now the server wants to send a push notification, it sends the content of the notification together with the device token to the APNs, which validates the push request, identifies the device corresponding to the token and sends the notification.

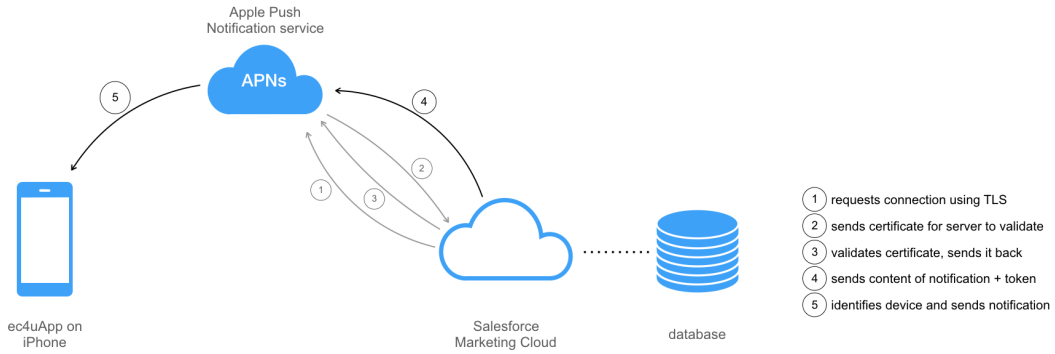


Figure 2: Sending a push notification over the APNs

4.2.2 Salesforce MarketingCloud

The other already existing system, that the app will be integrated in, is the Salesforce Marketing Cloud. As described in chapter 3.1, the Salesforce Marketing Cloud is a platform for performing marketing automation activities based on and using stored customer data so that customers can be reached via different channels throughout their journey consistently. With the 'Journey Builder', a Marketing Cloud tool, an individual customer journey can be created, eg. one called "Abandoned Cart", which tries to re-activate customers who left a buying process without finishing their purchase. Within a journey, several touchpoints with the customer are defined and the channel through which the contact happens. A common channel is the email channel. In this case, the Marketing Cloud tool 'Email Studio' is used in combination with the Journey Builder to determine for each touch point a suitable email. Integrating the app means that not only emails will be sent throughout the journey, but also push notifications. The Marketing Cloud tool for push notifications is MobilePush, which is part of the 'Mobile Studio'. Within the overall process of transmitting push notifications, as depicted in the previous chapter, the Marketing Cloud is the server and in MobilePush, the push notifications are created and sent off.

When generating a push notification in MobilePush, the desired type of notification is chosen from the following options: outbound, location entry, location exit, beacon and inbox. Outbound messages are sent based on a set time or demographic data, geographic messages are triggered by the customer entering or leaving a specified location and beacon messages depend on the customer's proximity to a physical beacon. Inbox notifications persist because they are saved in the app's inbox. After determining the type, the push notification is given a name, a title and subtitle and filled with content. This can be text only or text and media. Then, a send method is selected, depending on the chosen type of notification. In case of an outbound message for example, a time has to be set, at which the message will be sent. Recalling figure 2, these steps describe the first part of sending a push notification, which is generating it. Before this push notification can show up on the screen of the user's iPhone, a link has to be created between the Marketing Cloud and the app on the device. This is also handled by MobilePush. The app is added in MobilePush together with its certificates that enable a secure connection and were created for

the app in the Apple Developer Account. Subsequently, MobilePush outputs an access token, which is then supposed to be set in the app's source code. The access token is part of the settings that configure the Marketing Cloud SDK. The Marketing Cloud SDK should be integrated in the app to facilitate the connection between app and Marketing Cloud. This connection and the SDK's methods are used for it, is described in more detail in chapter 6.2.

4.3 Proposed System

Having introduced the systems that the app will be connected to, now the focus is put on the app. In the following, the requirements that the app should fulfill will be determined.

4.3.1 Functional Requirements

FR 1: allow push notifications
After installing and first launching the app, users can allow or deny push notifications. In case of denial, users will be asked again to allow push notifications at a later point.

FR 2: receive push notifications
Users, who chose to allow push notifications, receive them depending on:

- FR 2.1: their activities
- FR 2.2: time
- FR 2.3: updates
- FR 2.4: their location

FR 3: see saved push notifications in inbox
Push notifications containing important information are saved in the inbox, so that users can re-read them.

FR 4: enter login data
Upon first launching the app, users go through a login process, where they are asked to enter their name and email address and to set preferences. Users must login to use the app. Invalid user input is pointed out by the app and users get information on how to correct it.

FR 5: update profile
Users can view their profile and update it by uploading a profile picture or changing the settings.

FR 6: browse events
Users can look through events of different categories: events, they registered for, upcoming events or events close to them. Users can view detailed information about an event and mark it as favorite.

FR 7: check-in at the event
Users can check-in at the event via a QR-code.

FR 8: add events to calendar
Users can add events in their calendar app.

FR 9: browse news
Users can view the latest news related to the company. For detailed information they are directed to the

corresponding website (without actually leaving the app).

4.3.2 Non-functional Requirements

NFR 1: Usability
A pleasant user experience via an intuitive and simple interactions is important for the app because it is part of the overall experience a customer (the app-user) has with the company (the app-provider).

NFR 2: Utility
A responsive user interface supports the user in case they enter invalid input.

NFR 3: Integration
The app should integrate the Salesforce Marketing Cloud SDK and connect to the Marketing Cloud. In the Marketing Cloud, secure communication is established via a SSL certificate, which is why the app has to support it. The app needs internet connection. User input should be coupled to existing data in the Marketing Cloud and this way the app should be able to distinguish known from unknown users (who do not exist in the Marketing Cloud yet).

NFR 4: Implementation
The app has to be developed in Swift. To be able to receive push notifications, the app has to have push notification capability. This feature can only be enabled for developers with an Apple Developer Account.
Additional constraints:
XCode 10, iOS 13
In the Marketing Cloud, the maximum payload size of push notifications for iOS is 2048 bytes.

NFR 5: Supportability
The app should be integrated into the existing Salesforce Marketing Cloud instance of the ec4u to be able to ntegrate the app in various marketing automation activities performed in this instance (eg via Journey Builder, Email Studio and other Marketing Cloud products).

NFR 6: Reliability
The app should handle user input securely, so that it cannot be accessed by third parties.

NFR 7: Performance
The app should process user input and respond to interactions within less than 5 seconds.

NFR 8: Conformance to guidelines
The design of the app should conform to the company's branding.

NFR 9: Packaging
The app should be distributed via a download link.

4.4 System Models

4.4.1 Overview

Based on what the App should do, as specified by the requirements in the previous chapter, this chapter will explain, what the app can be used for. Furthermore the app will be modelled in different ways.

As described in chapter 3.2, push notifications are applicable especially in later phases of the customer lifecycle, when it is about maintaining the relationship to the customer. One way of doing so is by organizing an event for customers. Inviting the customers to the event and hosting it can be considered as a way of re-engaging with the customer, which is a common activity in the post-purchase phase. This example of event management is the basis to the analysis and modelling in the following chapters. First, a scenario is presented, describing a procedure one specific customer might go through while using the app. From this scenario, more general use cases are derived in the next chapter. The use cases are meant to identify actors and their relationship to the system. Then, according to the determined requirements and use cases, the app will be represented by different models.

4.4.2 Scenarios

The following scenarios are meant to exemplify how the app could be used in reality.

Scenario 1: Caroline just moved into her first own apartment. She ordered all of her furniture online from a company called FurniFla. A couple of days later, Caroline receives an email from FurniFla inviting her to their Opening of a new branch store, where the new Spring Collection will be exhibited and visitors get 10 percent off. Caroline likes this event and registers to it via a link in the email. The email also contains a link to download the new FurniFla App. Caroline clicks on it and the app installs on her iPhone. She opens the app and gets asked to allow the app to send her push notifications and to use her location - Caroline agrees with both. After logging in to the app with her email address, she starts browsing through upcoming events organised by FurniFla. A notification pops up offering her a quick tour through the app. Caroline follows it and then goes to her profile to upload a profile picture, as described in the tour. After that she returns to browsing through upcoming events. She clicks on the one she just registered for via email and gets more information on the event.

Scenario 2: A couple of days later, Caroline receives a push notification by the FurniFla app reminding her of the Opening event. She realizes that she completely forgot about the event, clicks on the notification and gets directed to a page in the app showing infos on the event, like day, time and location. She remembers that the event will take place tomorrow evening. She clicks on the address, which leads her to google maps, where she can plan how to get there. After that she return back to the FurniFla app and while looking through it, she realizes that she has already received a push notification two days, which she must have overseen. Since the notification got saved to her inbox, she can still view it.

Scenario 3: The next day, on her way to the event, Caroline missed the ubahn, which is why she has to hurry to get there. While she is walking from the ubahn station to the new FurniFla store, she receives a push notification welcoming her to the event and letting her know that she can check in at the event with a qr-code, which will show up just by clicking the notification. Caroline is glad she does not have to lose time by looking for her registration confirmation, opens the notification and the qr-code is displayed. Her qr-code gets scanned and she can enter the event.

Scenario 4: After exploring the new furniture collection and buying some accessories for her kitchen,

Caroline goes home. On her way, she receives a push notification saying that it was nice that she joined the event and letting her know that she can check which events are coming up next in the app. After she clicked the notification, she gets to the overview of events and browses it.

4.4.3 Use Cases

The previously described scenarios will now be generalized as use cases. An overview of all use cases is illustrated by the use case diagram in figure 3. A use case diagram identifies actors and systems and how they interact with each other. In this case, there is only one actor, the Customer, who uses the App, and two systems, the App and the Marketing Cloud. The Customer uses the different functionalities of the App and the App and the Marketing Cloud are connected to each other to exchange data. If the Customer logs into the App (FR4), the App establishes a connection to the Marketing Cloud to forward the login data. In the Marketing Cloud, a contact is created corresponding to the login data. Having logged into the App, the Customer can browse through upcoming events (FR6) that the App displays, after retrieving the corresponding data from the internet. The Customer can also view details on a specific event after clicking on it. Upon receiving a push notification (FR2), the Customer can open it and gets directed to the detailed view of an event or to the inbox, depending on the content of the push notification. The content and type of the push notification is determined in the Marketing Cloud. The types that can be chosen from include time- and location sensitive, Inbox and In-App push notifications. After sending off the push notification in the Marketing Cloud, the App is responsible for displaying it correctly and handling how the Customer can interact with it. However, the App can only receive the push notification, if a connection to the Marketing Cloud exists and if it has access to the Apple Push Notification service (APNs).

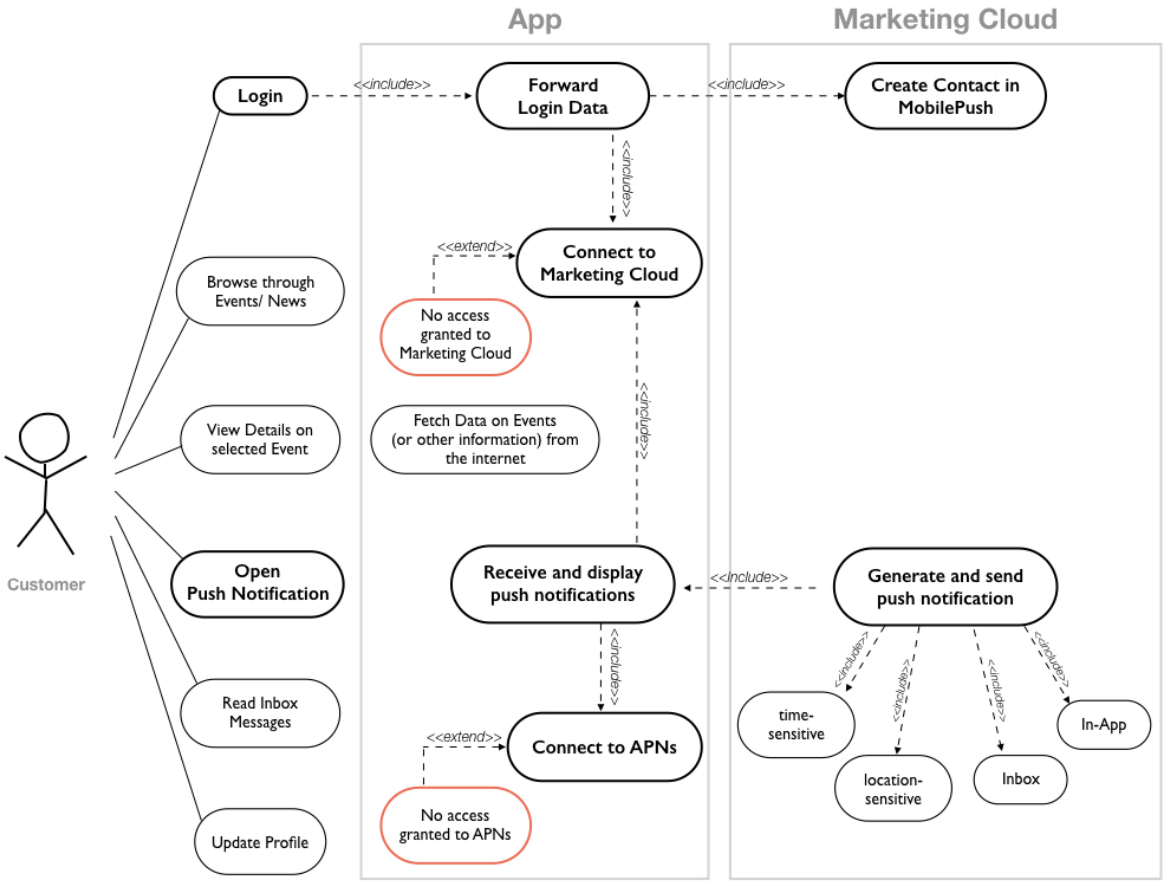


Figure 3: Use case diagram

The login process plays a crucial role because it determines whether user input and existing data are coupled correctly and whether the app can be assigned to a unique identity. Therefore, the use case "Login" is described in more detail in the use case table in figure 4.

Use case name	**Login**
Participating actors	initiated by: Customer communicates with: App Marketing and Cloud
Flow of events	1. The Customer chooses to allow push notifications, enters login data and clicks on the login button. 2. The App saves the login data to the App's documents directory. 3. The App sends the login data to the Marketing Cloud. 3.1 The App requests a token from the Marketing Cloud to be able to access it. 3.2 The Marketing Cloud sends a token 3.3 The App sends the token + the email address (from the login data) to the Marketing Cloud to request the corresponding key. 3.4 The Marketing Cloud sends the key. 3.5 The App enables the Customer's registration in the Marketing Cloud by sending the Customer's key to the Marketing Cloud. 3.6 The Marketing Cloud updates the corresponding data entry under which the Customer exists in the Marketing Cloud. 3.7 The App displays the Home Screen. 4. The Customer has access to the content of the App.
Entry condition	A data entry of the Customer already exists in the Marketing Cloud. The Customer has installed the App and allowed push notifications. The Customer enters all login data correctly. The App has access to the internet.
Exit condition	The Customer has access to the content of the App and can use it. The Customer's data entry in the Marketing Cloud was updated so that it includes that the Customer is an App user.

Figure 4: The use case "Login"

For this project, the most relevant use case of the app is to receive push notifications. This use case can be further split into more detailed use cases, each describing a situation in which one specific type of push notification is useful. Such use cases are presented in the following, based on the context of event management. Different types of push notifications are applicable in different phases of the event management. For example, before the event takes place, a time-sensitive push notification can be useful to remind the user of the event. Accordingly, push notifications can be applied in different ways throughout the management of an event, which results in several use cases for push notifications. They are summarized in figure 5, based on the scenarios described in chapter 4.4.2. To create a better understanding of these use cases, one of them, the "Registration Confirmation", is described in more detail in figure 6.

	Pre-Event			Event		Post-Event
	Scenario 1	Scenario 2		Scenario 3		Scenario 4
use case	Onboarding	Registration Confirmation	Reminder	Welcome/ Check-In	Info/ Update	Goodbye
PN type	In-App	Inbox	time sensitive	location sensitive	location or time sensitive	location sensitive

Figure 5: Overview of use cases of push notifications

Use case name	**Send Registration Confirmation**
Participating actors	initiated by: Marketing Cloud communicates with: Customer and App
Flow of events	1. The Marketing Cloud recognizes that a Customer who registered for an event, installed the App and logged in. Therefore, the Marketing Cloud sends the confirmation that the Customer is registered for this event, as an inbox message to the Customer. 2. The Customer receives the inbox message as push notification. By opening it, the App opens and displays the inbox (if the Customer does not open the push notification, a badge icon indicates that there is an unread inbox message).
Entry condition	The Customer has registered to the event via a registration link sent per email. The Customer has installed the app and went through the initial login process. The Customer allowed push notifications.
Exit condition	The Customer knows that his or her registration to the event was successful and can access the confirmation, including information on the event, any time.

Figure 6: The use case "Registration Confirmation"

4.4.4 Analysis Object Model

In this chapter, all objects used by the app will be presented and modelled in UML. As stated before, the central functionality of the app for this project is to receive push notifications. Subsequently, the app needs to handle a PushNotification object. A push notification is generated in MobilePush, which is why all classes related to the PushNotification class are colored grey. MobilePush offers different types of push notifications, which are represented by the subclasses Outbound, Inbox and GeoLocation. A PushNotification object is filled with PushNotificationContent, which includes among other properties the actual message of the push notification. In the app, an incoming push notification is handled by the AppDelegate class. This class also calls the methods that prepare the app for being ready to handle a

push notification. Among these methods, there is the registerForRemoteNotification() function, which registers the app with the APN server so that the app is eligible to receive push notifications, and the requestAuthorization() function which prompts a pop-up window asking the user for permission to send push notifications. Also part of the preparatory methods are the Marketing Cloud SDK functions which build and configure the SDK. After the set-up is done, the app can receive a push notification. When a push notification arrived, the app call the sfmc-setNotificationUserInfo(userInfo) function, which extracts the message, which is the userInfo, from the PushNotification object and displays it accordingly on the device screen. The sfmc-setNotificationRequest(request) function handles the interaction of the user with the push notification. The corresponding object model can be seen in figure 7.

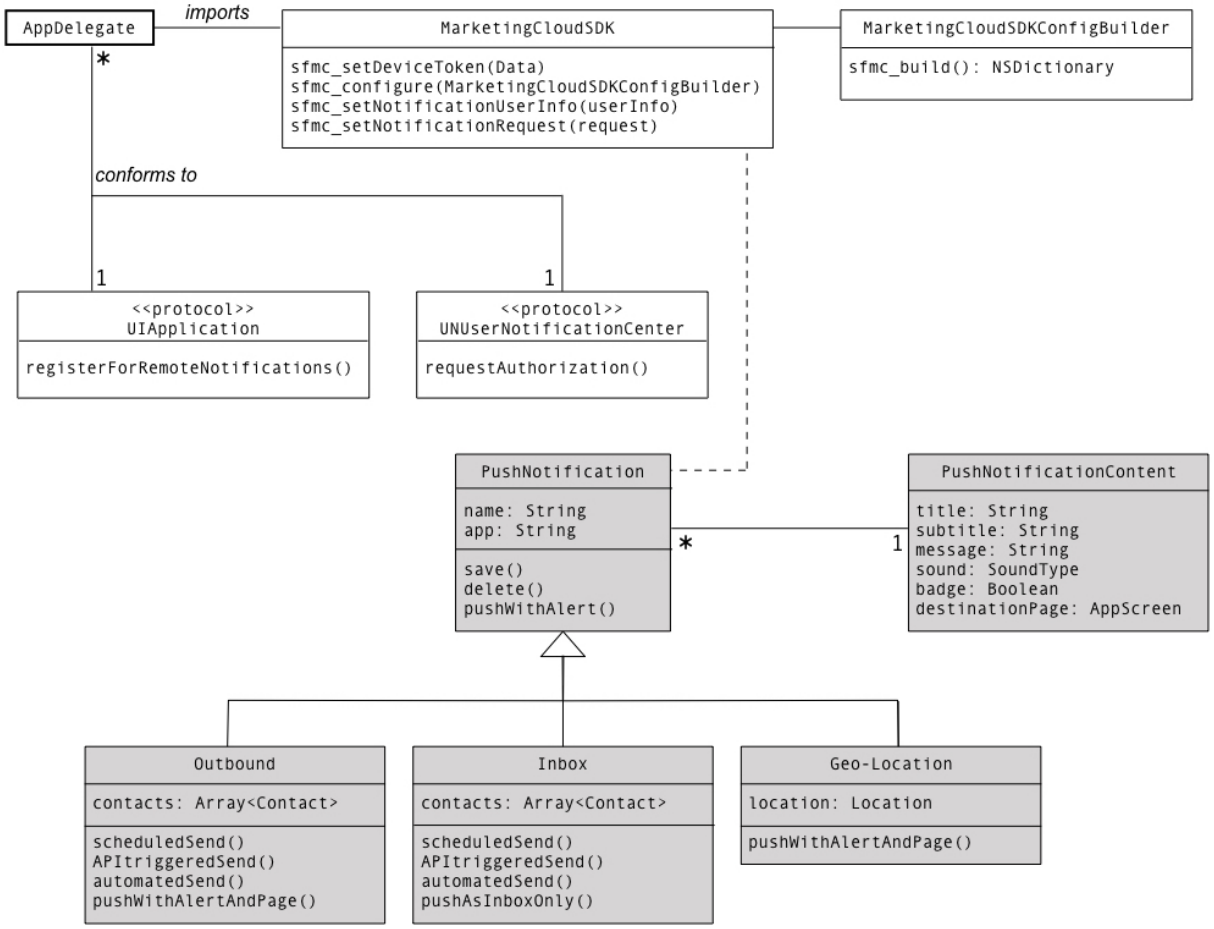


Figure 7: The AppDelegate class and the PushNotification class in UML

Another important part of the app is its login functionality, which is necessary for the push notifications to work because otherwise the device on which the app is running cannot be associated with a user. Subsequently, the login data has to be linked to the data entry in the MobilePush so that MobilePush can identify the app, that it should sends its push notifications to. The property that links a LoginData object to the MobilePushContact object is the email address. A LoginData object gets its values from the LoginScreen, which takes in the user input via its text fields. The LoginScreen class also includes methods for coupling the LoginData object to the MobilePushContact object. As explained later in more detail, the app cannot simply forward the login data to the Marketing Cloud, but first has to request a token by calling the requestMCToken() function to be able to access the Marketing Cloud. Before it can update the MobilePush contact with the login data, the app has to request the contact key from the Marketing Cloud by calling the requestContactKey() method. Then, the login data is linked to the

MobilePush contact by setting the key calling the sfmc-setContactKey(String) function. The UML model of the login process is shown in figure 8.

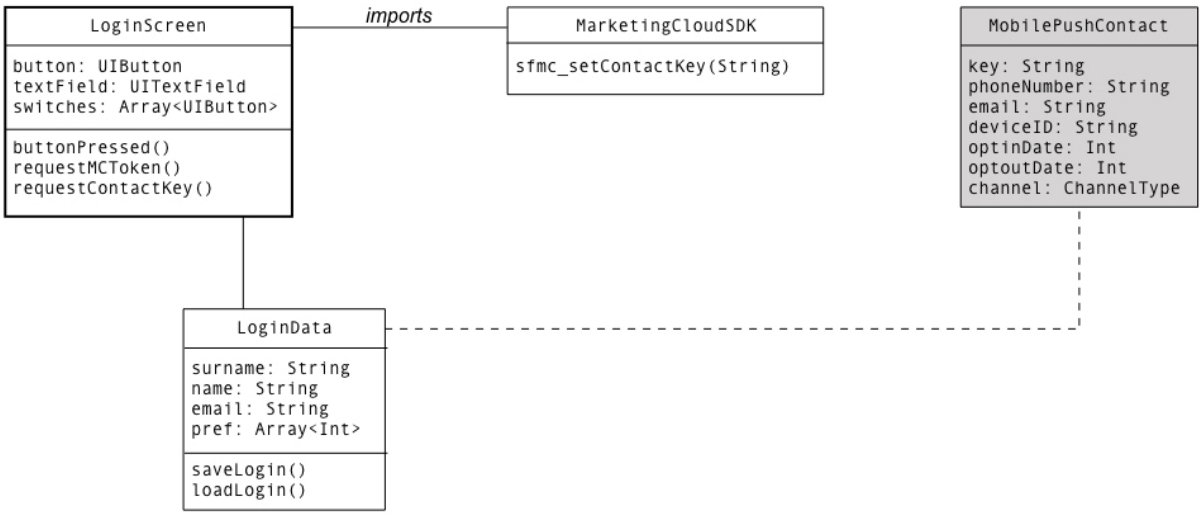


Figure 8: The LoginScreen class in UML

Once the user has logged in to the app, the app's content is shown. The first screen displayed after the login process is the HomeScreen. It has a label as property whose value is the user's name which is retrieved by loading the name property of the saved LoginData object. The HomeScreen displays its content in UICollectionsViews. A UICollectionView consists of multiple UICollectionViewCells. A cell cannot exist without a view, which is why this association is a composition. A cell has two subclasses which are adapted to the content object they hold. The content can be an Event object or a News object. The content objects are assigned to the cells by calling the fillCellsWithContent(Content) function which returns the filled cell. All objects included in the HomeScreen are shown in figure 9.

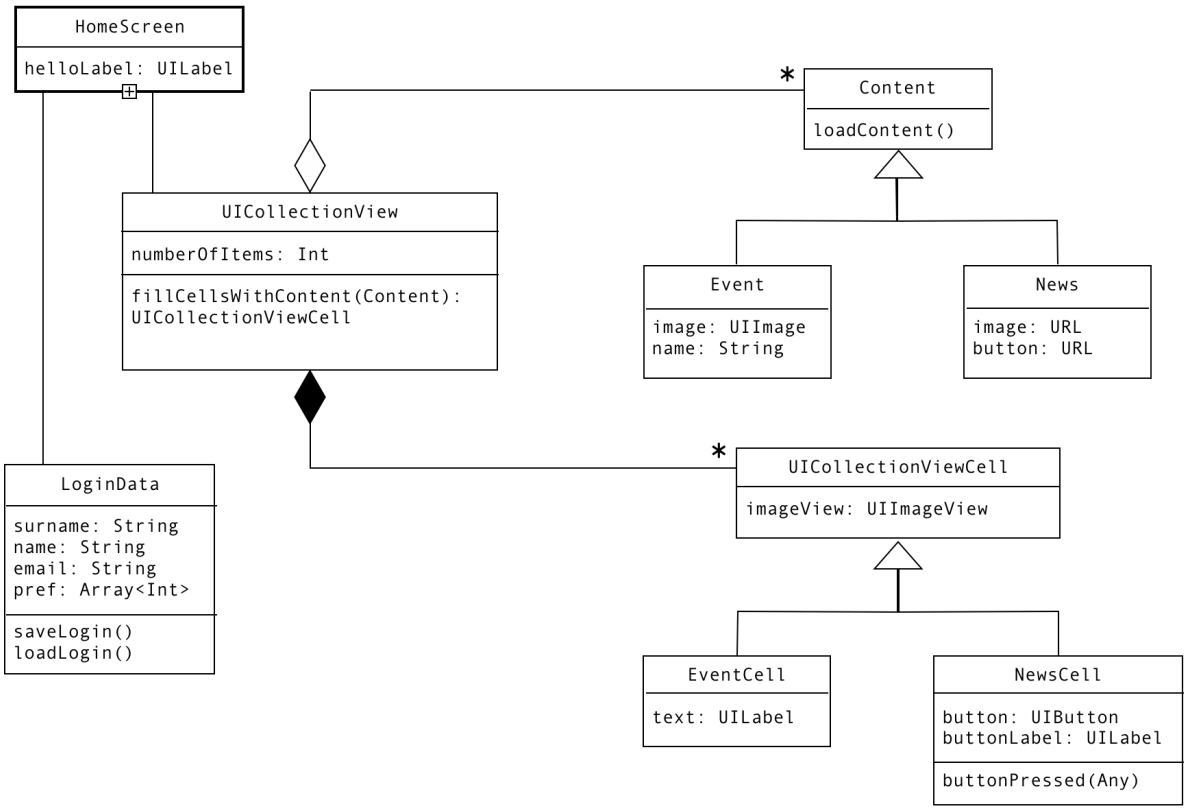


Figure 9: The HomeScreen class in UML

4.4.5 Dynamic Model

In this chapter, the defined requirements and use cases will be turned into a user flow. A user flow describes the different steps a user is going through when using the app. Prior to any other step is the user launching the app. The next step depends on the login status of the user: if the user has just installed the app and launches it for the first time, they have to go through the initial login process. If the user opens the app repeatedly, they either directly see the app's home screen if in their previous session they did not log out or they have to login again. However, since they have already gone through the initial login process, the repeated login is shorter and only requires the user to enter their name and email. In the initial login process additional information is asked, like the user's preferences. Eventually, the user reaches the home screen and from there has several options: They can navigate to another screen in the app displaying the inbox or their profile or they can click on content shown in the home screen. If they click on an event they get more details on this event or if they click on a news post they get taken to the website containing the corresponding blog post. Any of these options include the possibility to go back to the home screen. From the profile screen the user can also get to the login screen if they logged out. The user flow through the app is visualized as activity diagram in the following figure.

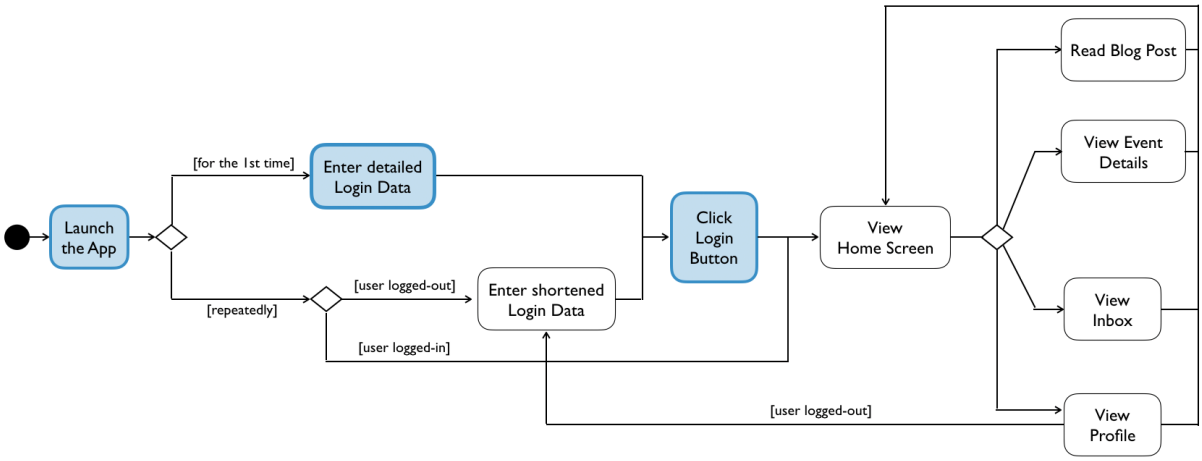


Figure 10: The user flow through the app

In the activity diagram in figure 10, three steps are highlighted in blue. They are part of the initial login process, that the user goes through after launching the app for the first time. During this process a lot happens in the background because this is when the connection to the Marketing Cloud is established and the entered login data is forwarded so that the corresponding contact in MobilePush is created and updated. These steps, which happen while the user goes through the initial login process are visualized in a detailed version of the previous activity diagram in figure 11. Actions performed by the user are colored in blue, actions performed by the app have a white background and everything that happens in the Marketing Cloud is grey.

After the user has launched the app, the app configures the Marketing Cloud SDK which involves asking for permission to send push notifications, requesting a device token from the APN server and sending it to the Marketing Cloud. At the end of the configuration, a MobilePush contact is created containing a random key, since the user has not yet entered any login data, and a set opt-in or opt-out date depending on whether the user has given permission to send push notifications or not. If the user enters their login data correctly, the login button is enabled and they can click it. With the button being pressed, the login data is saved and forwarded to the Marketing Cloud. In order to forward it, the app executes several steps. It first requests a token from the Marketing Cloud to be able to access it and then asks for the contact key that corresponds to the email address that the user entered. Depending on what the user entered and whether he or she is known (existent as contact in the Marketing Cloud), the Marketing Cloud returns the key associated with the user. The app then uses this key to link the login data to the MobilePush contact by replacing its random key and adding the user's email address.

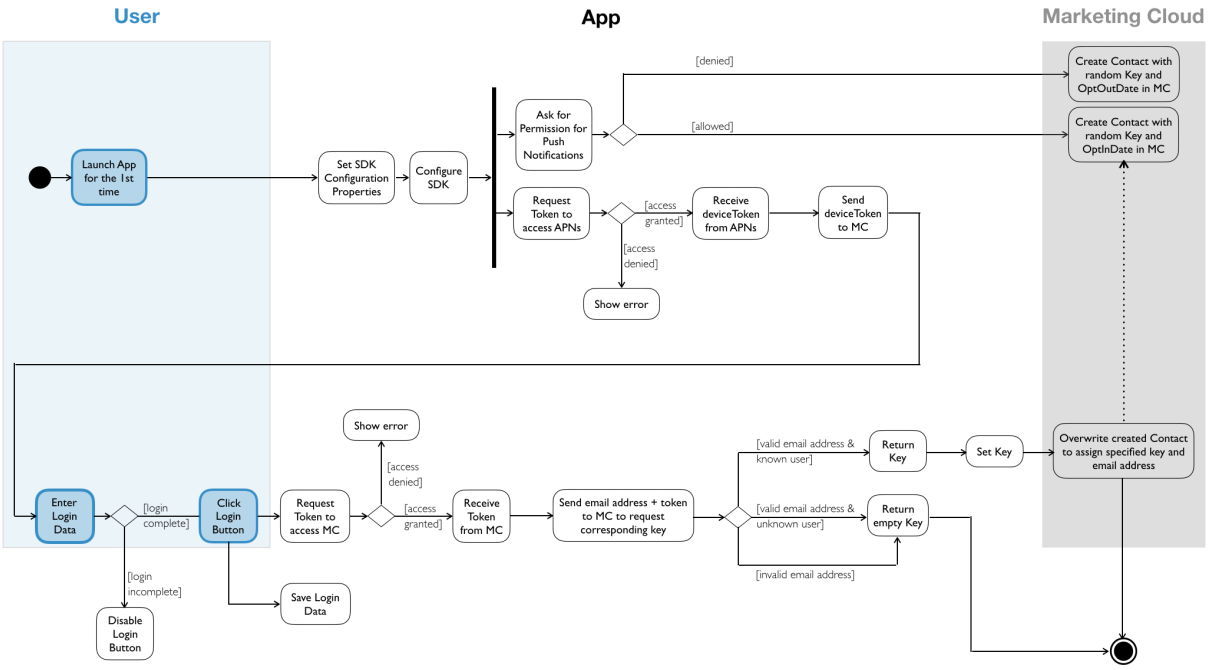


Figure 11: The user flow for the initial login process including all steps that the app and the Marketing Cloud execute in the background

4.4.6 User Interface

In this chapter, some of the screens of the app's mock-up are presented. The mockup was created according to the specified requirements and use cases.

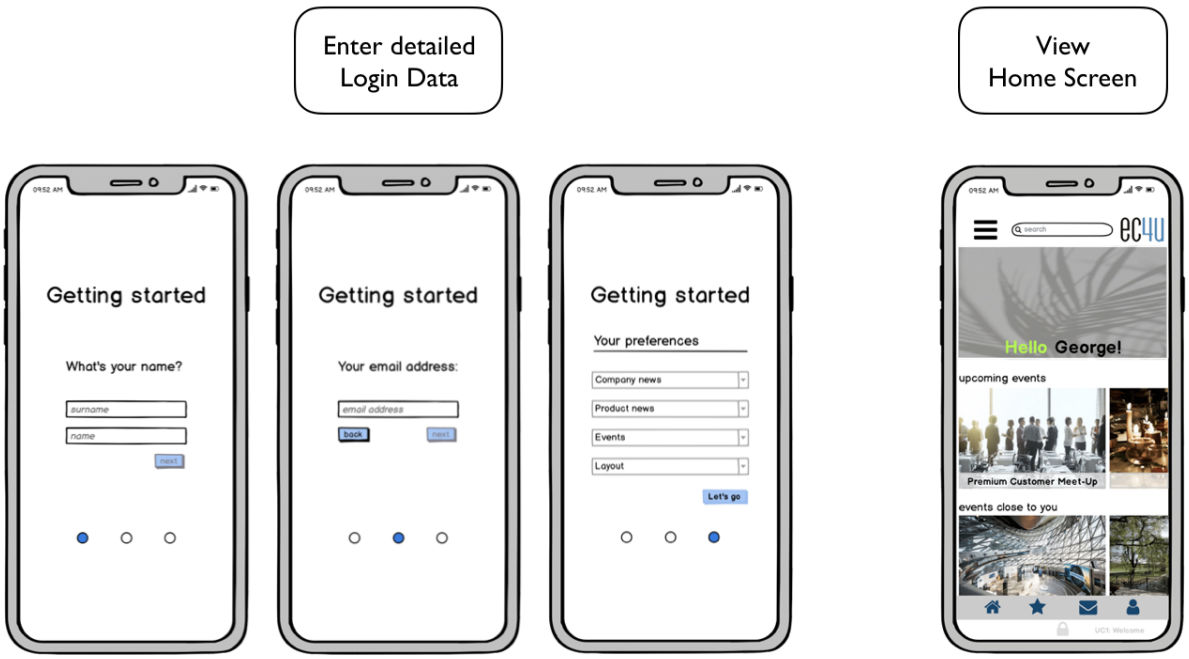


Figure 12: The Login Screens and Home Screen of the app as mockup

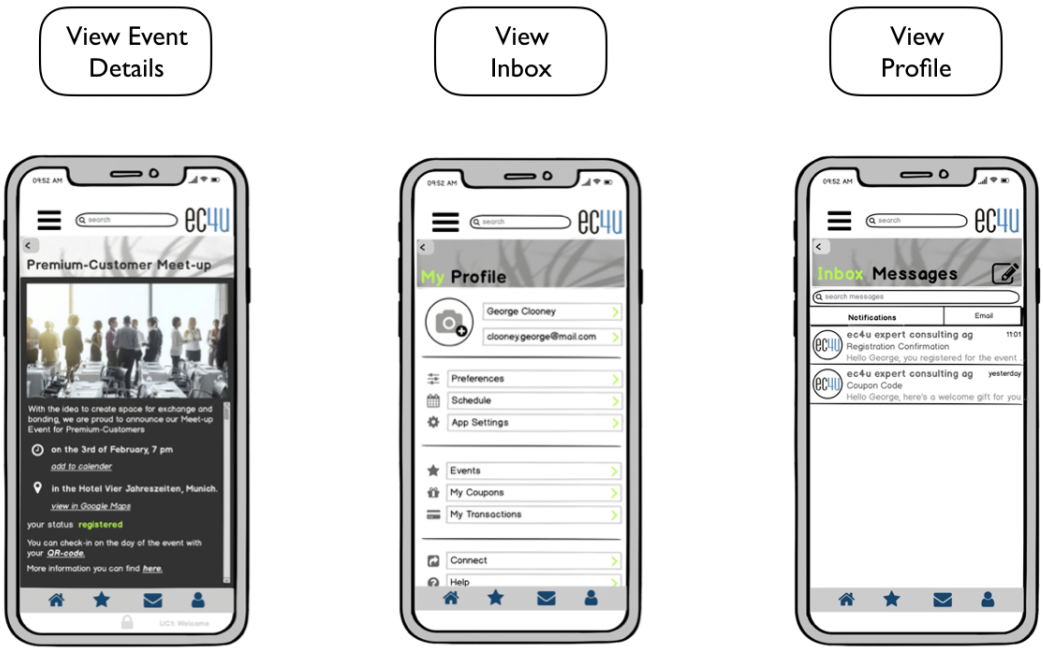


Figure 13: The Event Details, Inbox and Profile Screen of the app as mockup

	Pre-Event	Event	
use case	Onboarding	Welcome/ Check-In	Info/ Update
PN type	In-App	location sensitive	location or time sensitive

Figure 14: Screens of the app displaying an in-app message, the qr-code to check-in and the event's schedule as mockup

5 System Design

While the previous chapter described the app from a more theoretical point of view, what it is supposed to do and what it should be used for, this chapter explains how the app is designed so that these specifications can be realized.

5.1 Overview

As this chapter is determines how the app will be implemented, the app is no longer an abstract concept, but a concrete product. This is why, from this chapter on, the app will be called ec4uApp since it is developed for the ec4u expert consulting ag the practical partner of this project.

5.2 Design Goals

Before going into detail about the design of the app, the design goals will be determined and prioritized. This is important because design goals can repel themselves, for example a system that is safe, fast and cheap is not realistic. Design goals are derived from the non-functional requirements or are given by the ec4u.

Supportability
The ec4uApp is one tool among others used to communicate with the customers throughout their journey. Therefore, the ec4uApp has to be compatible with existing software, the Marketing Cloud, to seamlessly cooperate with it (NFR4). Consistency between different tools is one of the most crucial concepts in CRM, which is why this design goal has a high priority.

Extensibility/ Modifiability
Since the ec4uApp is part of a bigger project, it is likely to be used by different actors, who might want to extend or change its functionality to adjust it to their use case. Therefore, the ec4uApp should be easy to modify. This design goal refers to the future use of the app, which is why for this project, it has medium priority.

Documentation for open source development
This design goal comes along with the previous one: to make the ec4uApp extensible and modifiable, its source code and architecture should be well documented. A good documentation should be given under any circumstances, whether the app will be re-used or not. Therefore the priority of this goal is high.

Security
As stated in NFR5, the ec4uApp has to handle user input securely. This is important because the user input includes personal data, such as the user's email address. The ec4uApp forwards the user input to the Marketing Cloud and also needs to access data in the Marketing Cloud. Therefore, the ec4uApp should guarantee a secure connection for exchanging data. This design goal is prioritized highly.

Robustness
Since the user enters data in the ec4uApp to log in, it has to be robust to invalid user input. The login process is a crucial functionality of the app and therefore this design goal has a high priority.

Response time
As specified by NFR6, the ec4uApp should be able to respond to and process user input quickly. A functionality of the app, for which time is critical, is the time-sensitive push notification. It is triggered by

a set time and its content was specifically designed for that time, which is why it is important that there is no major delay in the delivery of this notification. Other than this, time is not a critical factor for the app to work, which is why this goal has a medium priority.

Usability/ Utility
The Usability and Utility of the ec4uApp is important, because a good user experience influences the overal customer experience, which is crucial for a positive relationship between customer and company. As stated in NFR1, the app represents the company and therefore is part of the customer experience. This goal has a high priority.

5.3 Subsystem Decomposition

To be able to implement a system, its detailed architecture has to be known. The architecture of a system describes what components it is composed of and how they interact with each other. This chapter presents where in the architecture of the complete system the ec4uApp is placed as subsystem and then decomposes the ec4uApp into its subsystems.

The complete system consists of the ec4uApp and the Marketing Cloud. The app requires services provided by the Marketing Cloud, like generating push notifications. This structure can be modelled as client-server architecture, with the app on the client side and the Marketing Cloud on the server side. The Marketing Cloud includes the database that stores customer data. On the server side, there is also the APNs server providing Apple's push notification service and a web server via which the app accesses web content. The app on the client side is modelled as 3-layer architecture. These layers are non-hierarchical and open. The app logic handles the push notification functionality and the stored data is the login data. A simplified overview of the client-server structure is given by the UML component diagram in figure 15.

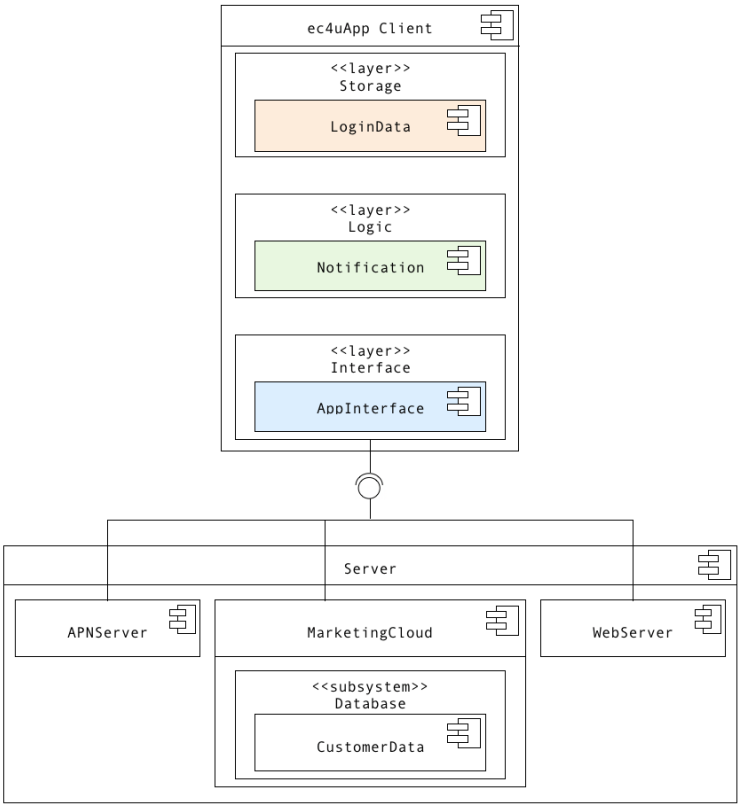


Figure 15: UML component diagram

In the following, each layer of the ec4uApp will be investigated in more closely. A detailed version of the UML component diagram is shown in figure 16. The app's interface communicates with the server-side and its components are the different screens of the app. The AppStateManagement is part of the logic layer and becomes actives as soon as the app is launched. It connects to the Apple Push Notification server to enable the app's push notification capability. It is enabled after the app has received a token via the Apple Push Notification service (APNs). Once the app is registered with Apple's server, the AppStateManagement component connects to the Marketing Cloud via the MarketingCloudSDK. The SDK includes all classes and functions the app needs to handle the push notifications sent by the Marketing Cloud. Consequently, the screens that include push notifications require the SDK, which is the Inbox screen and the Login screen. The Login Screen integrates the SDK because it needs to connect to the Marketing Cloud to request the key corresponding to the entered email address. The purpose of this is to create or update a data entry in the Marketing Cloud with the information, that a certain user is reachable via push notifications. This process is handled by the DataCoupling component, which is part of the app's logic layer. Using the input from the login screen and the information on whether the user chose to allow push notifications, it determines the connection between app and Marketing Cloud. In other words, it determines the data entry under which the user is saved in the Marketing Cloud. If all processes within the DataCoupling component are successful, the app is ready to receive push notifications. The app's response to receiving a push notification is handled again by the AppStateManagement component, i.e. how the notification is displayed, if a badge icon appears and what happens after the user has interacted with the notification.

Besides the services of the Marketing Cloud, components of the interface layer can also require services by other web servers that give them access to web content. This is the case for the home screen,

which displays data drawn from a website. The home screen also displays user data, like the user's name, to make the content more personal. For this it needs access to the LoginData, which is stored internally in the app's documents directory. This is represented by the storage layer. Data in the storage layer can also be accessed by other screen, like the Profile screen. The LoginData is retrieved from the Login screen, which collects and saves it.

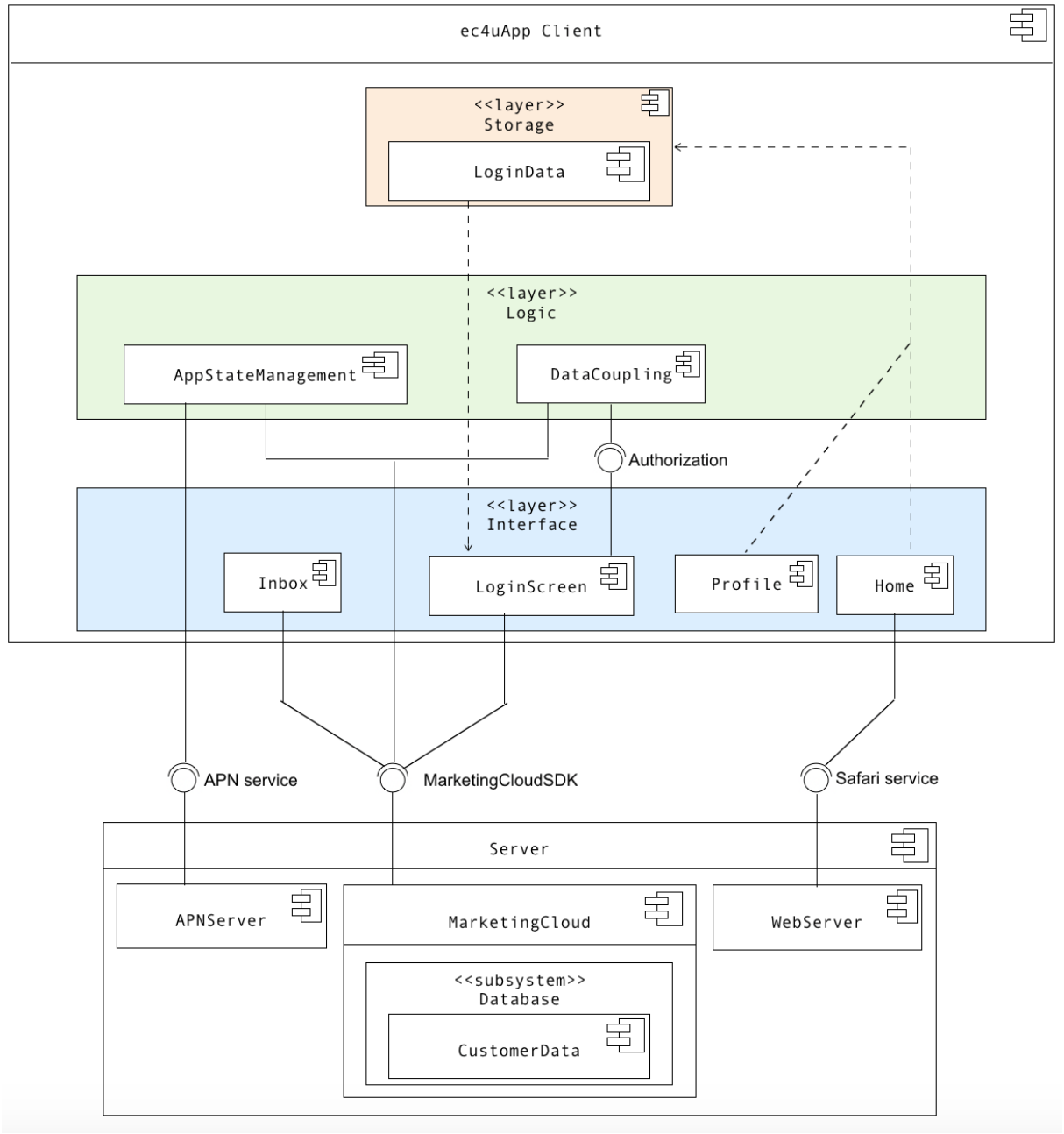


Figure 16: UML class diagram

5.4 Hardware Software Mapping

In this chapter, the previously identified components will be allocated in nodes. A node represents a specific piece of hardware. In this case, the hardware configuration is rather straightforward: all components included in the ec4uApp run on the user's device, which is an iPhone. The servers, that the

app communicates with, are different internet hosts. The allocation of components to nodes is shown in figure 17.

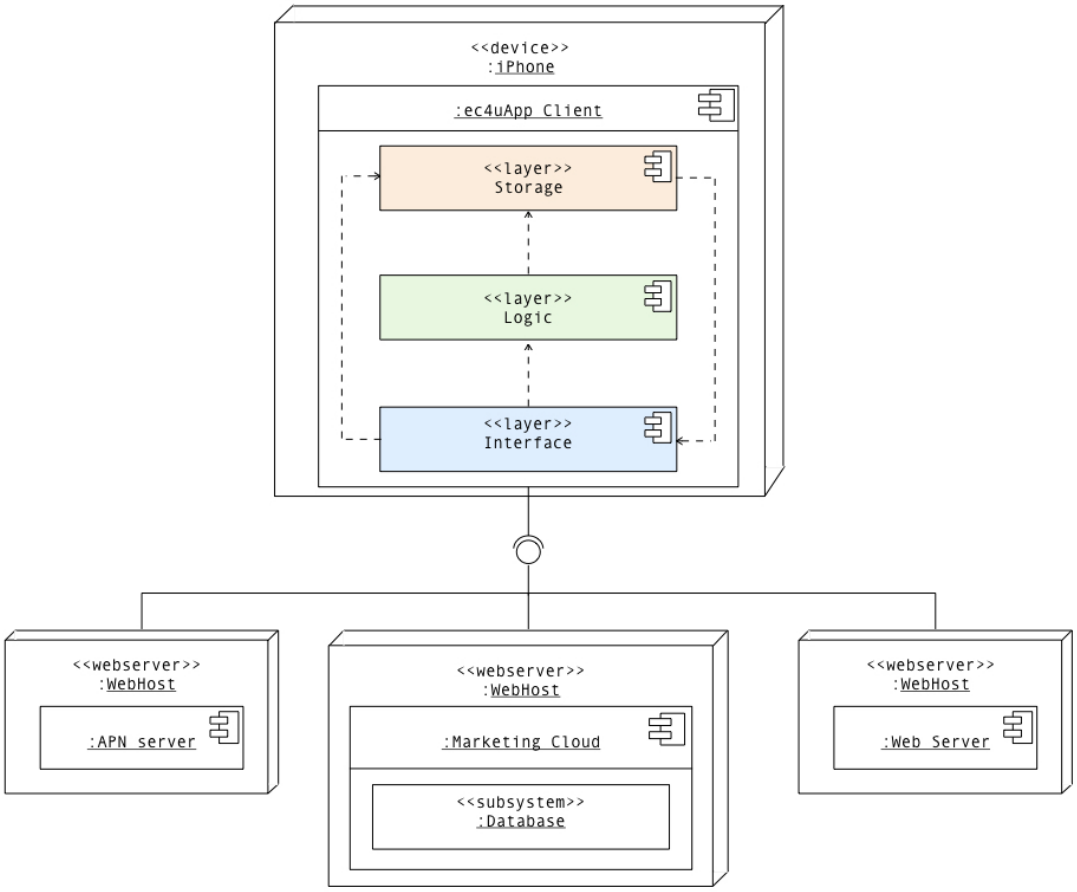


Figure 17: UML deployment diagram

Next, the communication between the hardware nodes will be investigated and how the services they provide or require are realized as software components. In order to trace the communication processes, participating objects have to be identified and allocated to their nodes. Classes involved in the communication between nodes are instantiated as shared singleton objects. The device node communicates with the server nodes either by calling functions on shared objects or by making http requests. Figure 18 shows how the objects are allocated in the components and which objects the components use to communicate with other components.

The communication between user device and servers can be split into three parts: preparing the setup for push notifications, which includes registering the app with the APN service and configuring the Marketing Cloud SDK, coupling data of the app with data in the Marketing Cloud and finally sending and receiving push notifications.

In the first part, the app communicates with the APN server to register the app for push notifications by calling the corresponding function on the singleton app object of the UIApplication class. The registration was successful if the app received a device token from the APN server. Then, a shared instance of the Marketing Cloud SDK is configured. The configuration causes a MobilePush contact to be created in the Marketing Cloud, which includes information that was set when the configuration of the SDK was

built, eg the opt-in date, if the user chose to allow push notifications. These communication processes originate from the AppStateManagement component in the logic layer of the app.

In the second part, the Marketing Cloud database is synchronized with the app's user input. For this, the DataCoupling component sends a http request to the Marketing Cloud. The request includes the email address, that the user entered, and asks for the corresponding key saved in the Marketing Cloud. After the app has received the key, it updates the previously created contact in the Marketing Cloud with the correct key by calling the appropriate SDK function.

For the third part, the sending and receiving of push notifications, a connection between Marketing Cloud and APN server has to be established because the delivery of the push notifications happens via the APN server. The Marketing Cloud generates the push notifications and forwards them to the APN server, which then determines the device they will be sent to. The connection between Marketing Cloud and APN server is established via a http protocol and secured by a SSL (Secure Socket Layer) certificate. The SSL certificate is created in the Apple Developer Account and assigned to the corresponding app. After the APN server has delivered the push notification to the app, the app has to do two things: it has to display its content and handle the user interaction with the push notification. This is taken care of by the AppStateManagement component, which calls the appropriate functions on the UIApplication, the UNUserNotificationCenter and the SDK object.

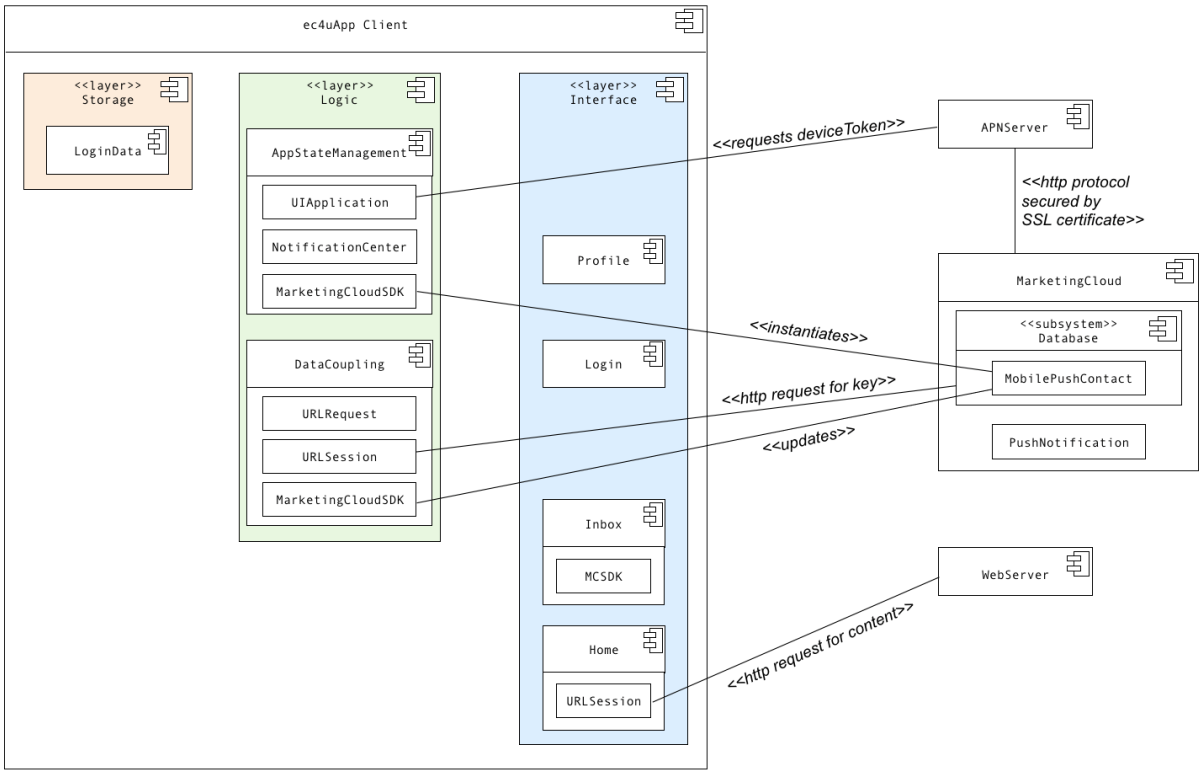


Figure 18: The communication between objects of different hardware nodes

5.5 Persistent Data Management

The ec4uApp contains different kinds of data. In the following, it will be determined which of this data should be persistent, or in other words, which data should have a lifetime across app sessions. To recall the app's content, its class diagrams are reviewed shown in figure 19.

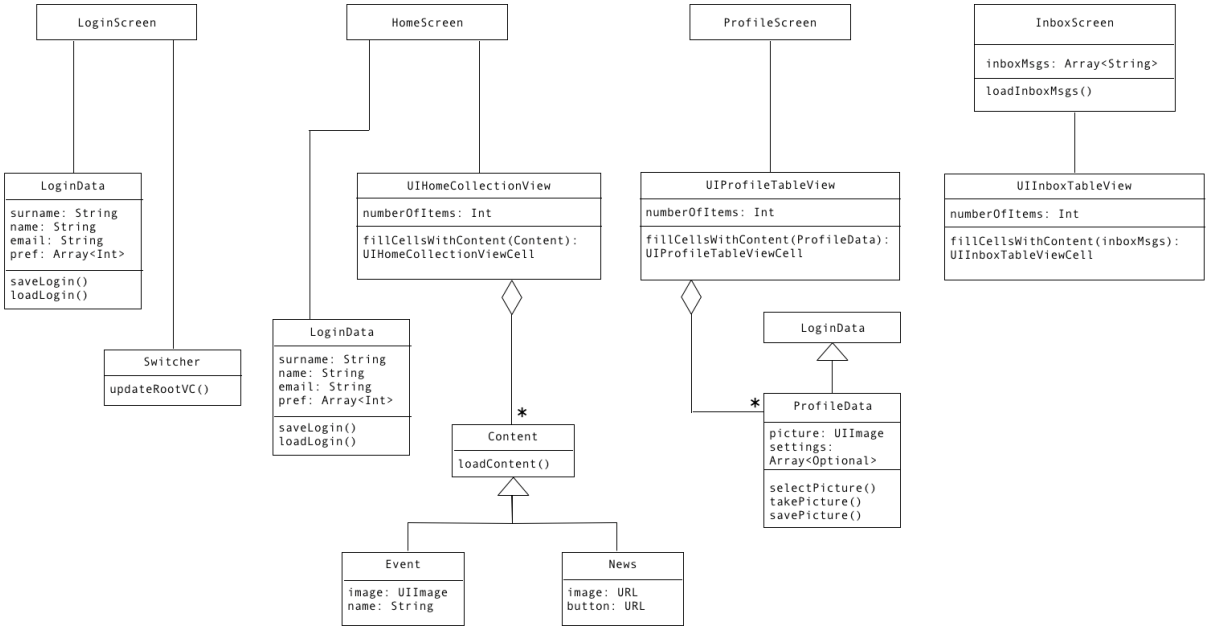


Figure 19: Data model

An obvious candidate for persistency is the login data. If the user reopends the app, data he or she entered in a previous app session should still exist. Therefore, the app has to store the user's login data and set preferences. Next to this, it also has to save the information that the user is logged in to display to correct screen after the user opens the app. The class representing this information is called Switcher. Consequently, the classes LoginData and Switcher, which are both associated with the LoginScreen, should be persistent. Data displayed by the HomeScreen, however, is dynamically downloads from a corresponding web server during run-time. This means, it does not have to be saved locally in the app and therefore is not persistent. Similarly, inbox messages displayed on the InboxScreen are not stored either because the MarketingCloudSDK provides a function that fetches all active inbox messages every time the user opens the inbox. Finally, data contained in the ProfileScreen clearly has to be persistent because the user does not want to setup its profile anew after re-opening the app. In summary, objects of the classes LoginData, Switcher and ProfileData have to be saved.

Knowing what data should be stored, the following will describe how it should be stored. In general, there are two strategies for managing saved data: as file system. or in a database. In order to select the right strategy, properties of the data to be saved will be examined and who will need to access it and how. The choice of the strategy also depends on the previously defined design goals of the system as defined in chapter 5.2.

Databases can be relational or object-oriented. Both are suitable for medium-size to large datasets and provide concurrency management. Object-oriented databases are especially useful for data including a lot of associations and relational databases enable accessing data via complex queries. The persistent data of the ec4uApp is small and simple because it does not contain many associations. Furthermore it neither has to be accessed via queries, nor by anyone other than the app, which is why concurrency is not an issue. These conditions indicate that the storage needs of the ec4uApp are best met by using a file system. The design goals support this choice: a crucial goal of the ec4uApp is to design it in a way that makes it compatible with the Marketing Cloud. The whole system should be consistent so that data can be exchanged easily. To achieve this, it should be possible to retrieve the stored data quickly. For this

purpose, file systems are more beneficial, since accessing data in databases may take more time. This is also in accordance with the design goal to guarantee a reasonable response time. Another design goal of the ec4uApp is to make it easily extensible or modifiable since it might be reused in a slightly different context. The less complex the data storage is managed the easier such changes can be made in the future, which also speaks for a file system.

In iOS, each app is provided with its own file system called sandbox. The sandbox contains different containers for different kinds of data as shown in figure 20 [1].

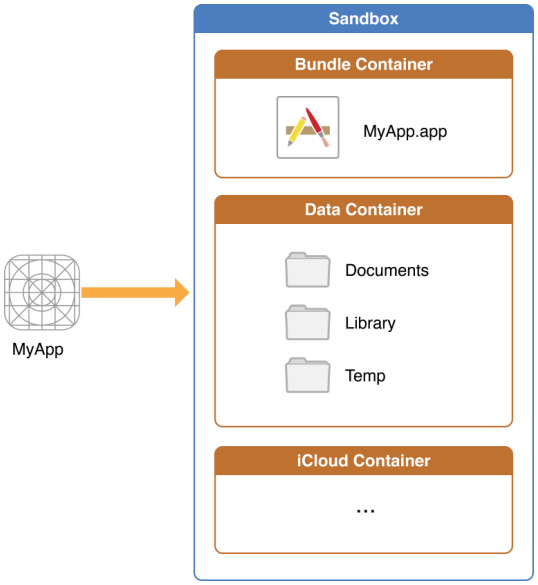


Figure 20: Structure of the iOS application sandbox

The Bundle Container does not allow writing data and the iCloud Container is accessible from multiple devices which is not desirable for login data. Hence, the suitable container for the persistent data of the ec4uApp is the Data Container and within this container the data is stored in the Documents folder. To save data to and load data from this folder, the LoginData and the ProfileData class adopt the NSCoding protocol. This protocol declares two methods for encoding and decoding object to archive and unarchive them. Using the NSCoding protocol, saving an object is implemented by assigning the value of each property of the object to a particular key. To load an object, the respective value associated with a certain key is determined. Apart from declaring methods for saving and loading the data, the path to the location in the Documents folder, where the data is saved, needs to be defined. For this, first the url for the Documents folder has to be determined and then a url has to be created specifically for the stored data. This is the way the LoginData and the ProfileData object is saved and loaded. An object of the Switcher class determines the screen the user sees first when opening the app. This information is part of the app's default system, which stores the user's preferences on the app's behavior. To interact with this system, the UserDefaults class can be used. Therefore, the screen that shows first after the app was opened is set and stored via a UserDefaults object.

[1] https://developer.apple.com/library/archive/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html

6 Implementation

6.1 ec4uApp Implementation

Immediately after the ec4uApp is opened, its launch screen is shown. The launch screen is displayed until the app has finished setting up and loading its initial content. During this process, the method of the AppDelegate class is called, which launches the app. The AppDelegate's methods are responsible for handling the app's state, which determine whether the app is active or terminated or in the foreground or background. The AppDelegate class manages the app's lifecycle and creates a window where its content is drawn and an entry point to it. The app's content is managed by ViewController classes. Every screen in the app has its own ViewController class. This class handles the data and UI elements the screen uses and includes methods for the transitioning between screens. The ViewController, that the entry point is set to, is the root ViewController and its corresponding screen is what is shown first, right after the launch screen. The RootViewController of the ec4uApp is determined by the Switcher class according to the login status: if the user opens the ec4uApp for the first time, the first screen of the initial login process is the root ViewController. If the user has already logged in in a previous app session, the home screen is the root ViewController and therefore shown directly after the launch screen. If the user has logged in and out in the previous app session, a single login screen is set as root ViewController, which allows the user to re-login.

Figure 21: Implemented login screens for the initial login process

The ViewController classes of the login screens, which can be seen in figure 21, assign the data that the user enters in the UI text fields to the corresponding properties of a LoginData object and save it. If the user clicks on the next-Button, a method in the ViewController class of the current login screen is called that triggers the following login screen to be displayed. The next-Button is only enabled, if the user has entered data in the text fields. After the button of the last login screen (or the only login screen, if the user re-logs in) is clicked, the home screen of the ec4uApp is shown. Its ViewController class includes a label welcoming the user by their name, a collection view and a table view. The label displays the user's name by loading the value of the name property of the saved LoginData object. The collection view contains content about upcoming events and the table view shows previews to blog posts on the ec4u website. To fill the views with content, the corresponding objects of the classes Event and News are used. The properties of these objects, an image and a text or button, are assigned to content fetched from the

internet. Then the cells of the collection and table view are filled with this content respectively. If the user taps the button below the blog preview a method is called that opens the web page including the complete blog post in Safari, but without leaving the app. It displays the page together with a navigation bar, that brings the user back to the home screen.

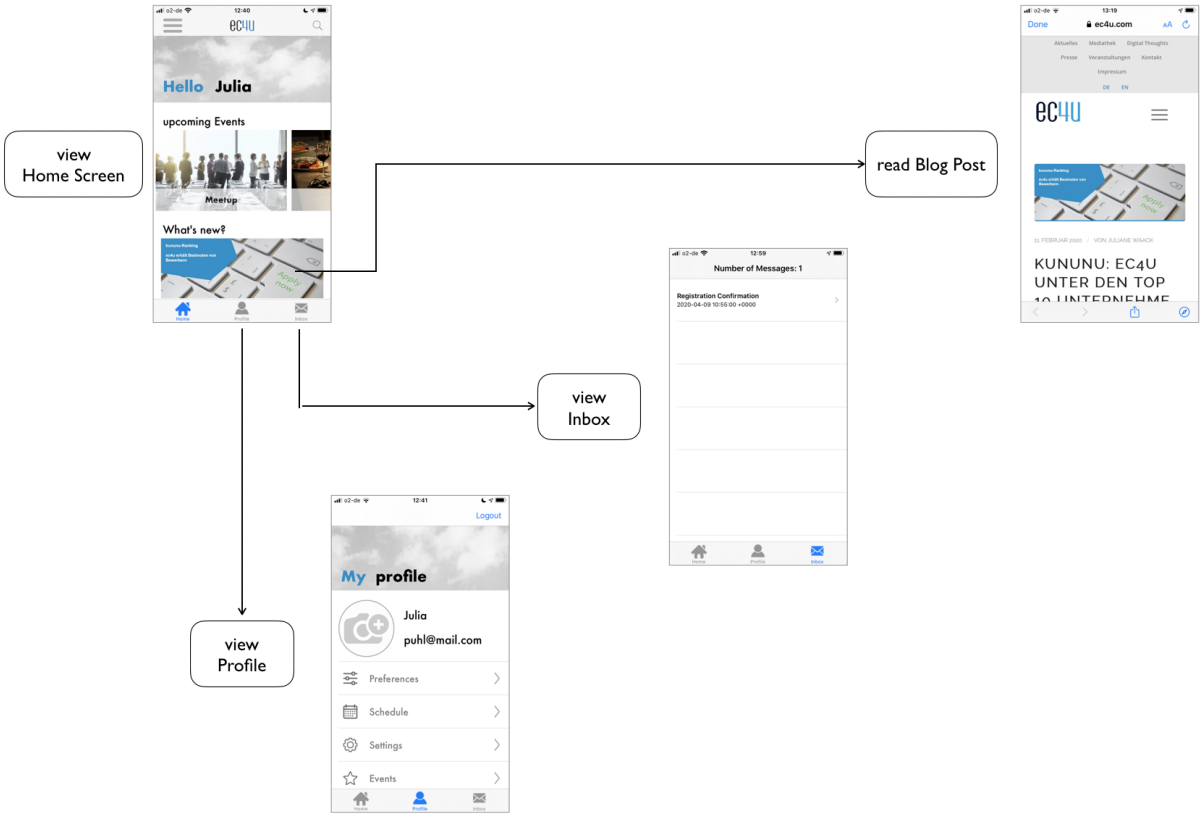


Figure 22: Implemented screens reachable from the home screen

The tab bar at the bottom of the home screen, as shown in figure 22, enables the user to navigate to the inbox and the profile. The navigational style of the ec4uApp is hierarchical meaning that the screen are stacked so that the user must retrace their steps to return to a certain screen. The navigation is managed by NavigationController classes. The login process and each tab of the home screen have their own NavigationController. The navigation between the tabs is handled by the TabBarController class. The navigation hierarchy of the ec4uApp is visualized in figure 23.

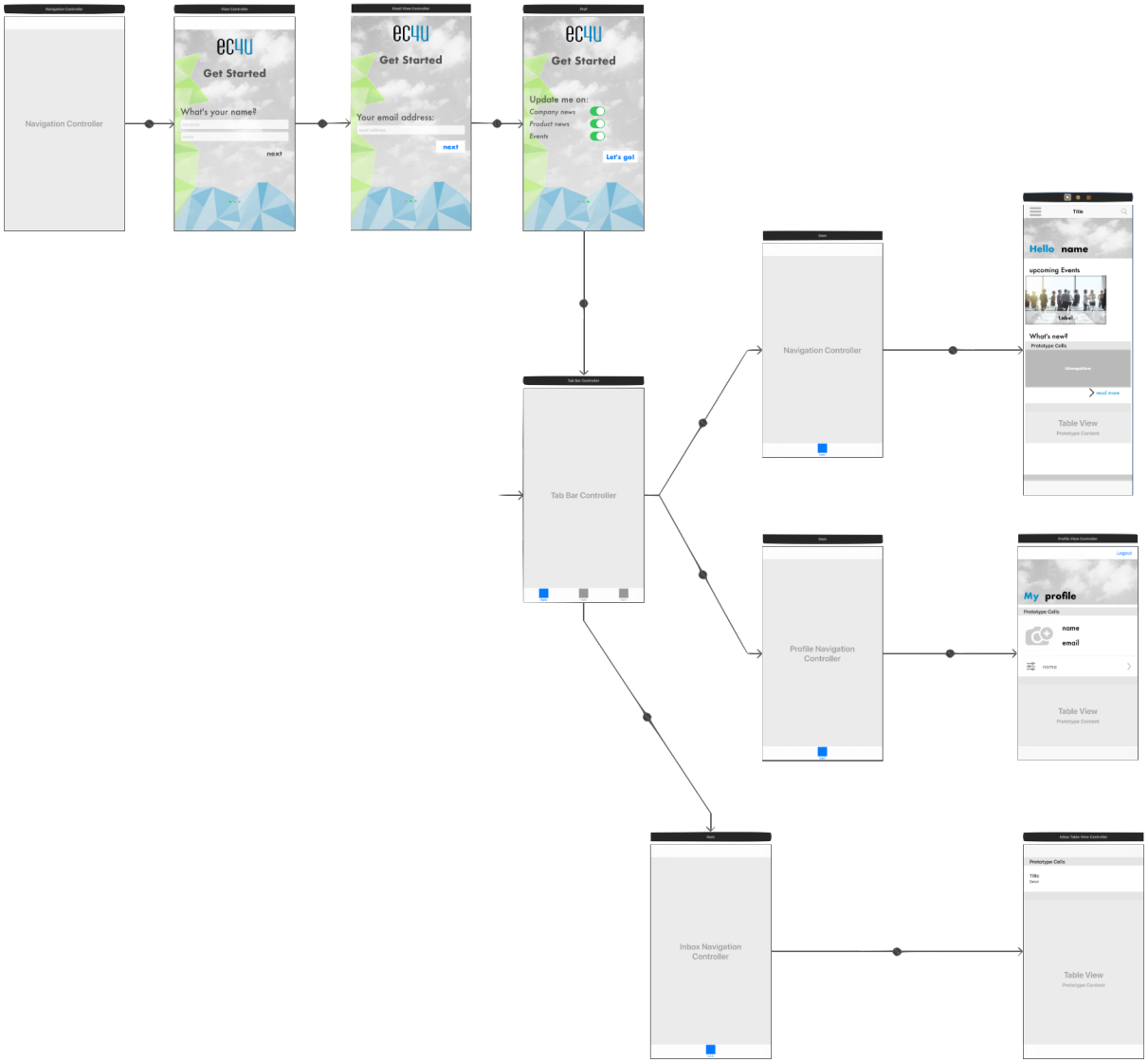


Figure 23: Navigation hierarchy of the ec4uApp

6.2 Connection between the ec4uApp and the Marketing Cloud

The development of the ec4uApp so far included filling it with content and implementing its internal functionality. One essential component is still missing and that is its capability to receive push notifications. For this, the ec4uApp has to communicate with two servers, the Marketing Cloud generating the push notifications and the Apple Push Notification (APN) server, which delivers the notification to the app. The steps required for setting up these connections are described in the following.

First, a SSL certificate has to be created, which is needed to secure the connection between the Marketing Cloud and the APN server. This is done in the Apple Developer Account by selecting the bundle ID of the ec4uApp, enabling its push notification capability and choosing to let a SSL certificate be generated for it. Before this process can be completed, a certificate signing request has to be created and uploaded to the Apple Developer Account. Then, the SSL certificate is available and can be downloaded. The ec4uApp is only actually capable of receiving push notifications, if this capability is also included in the app's 'Signing Capabilities' section of the app's target in Xcode and the developer team is set to the correct Apple Developer account.

To use the SSL certificate for the connection between Marketing Cloud and APN server, it is uploaded to the tool MobilePush. MobilePush then generates values for an access token, an app ID and app endpoint. To link these values to the ec4uApp, they are set in the app's code when configuring the Marketing Cloud SDK. Before the SDK can be configured, it has to be added in Xcode under 'Frameworks, Libraries and Embedded Content' in the target's 'General' settings. Then the Marketing Cloud SDK can be imported into the AppDelegate class. In this class, a method is declared, that configures the SDK with the values provided by MobilePush. The configure method uses functions of SDK's built-in classes. It includes a base class called MarketingCloudSDK and a class providing methods for setting the parameters of the configuration called MarketingCloudSDKConfigBuilder. Among these parameters, there is the app ID, access token and the app endpoint, which are assigned to the values provided by MobilePush. The settings of the SDK configuration also include information on whether the user opted to allow push notifications and whether they chose to let the app use the user's location, which is important for location-sensitive push notification. This way, when the configure method is called, a contact in MobilePush is created that contains a opt-in date or a opt-out date accordingly. At this point, however, the contact is not linked to a data entry, that may already exist for the respective user, which is why it contains a random key and does not have an email address. The process of configuring the SDK is summarized in the first column of figure 24.

Once the SDK is configured which means the app is registered with the Marketing Cloud, the app has to be connected to the APN Server. For this, the app receives a device token from the APN server, which indicates, that the app has registered with the APN server and is capable of receiving push notifications. Using this token, the APN server can identify the correct device, the push notification is supposed to be sent to. The app obtains the device token by calling the registerForRemoteNotifications function on the shared instance of the UIApplication class. Subsequently, the device token is forwarded to the Marketing Cloud by calling the corresponding SDK function. The process of registering the app with the APN server is summarized in the second column of figure 24.

The last step is to couple the login data entered in the app to the corresponding data entry in the Marketing Cloud. After the SDK configure function is called, a random data entry is created in the MobilePush. Once the user has logged in to the app, the login data is used to update this random data entry so that the random key is replaced by the user's key. This is what the SDK function setContactKey(key) does. However, this is only possible if the user already exists as contact in the Marketing Cloud. Therefore, before calling the setContactKey(key) function, the app sends the login data to the Marketing Cloud to check whether there is a corresponding contact. If there is, or in other words, if the user is known, the Marketing Cloud sends back the key for that user, which is then used to call the setContactKey(key) function. Finally, the contact in MobilePush contains the key and email address associated with the user, who logged in to the app. This process is summarized in the third column of figure 24.

Configure the SDK and Register with the MC	**Register with APNs**	**Couple data of app to data in MC**

(token obtained in MobilePush)
set token to build and configure SDK:
`.sfmc_setAccessToken(accessToken)`
→ `.sfmc_build()!`
→ `.sfmc_configure(with: builder)`
	request and receive token:
	`UIApplication.shared.registerForRemoteNotifications()`
	forward token to MC:
	`MarketingCloudSDK.sharedInstance().sfmc_setDeviceToken(deviceToken)`
		use contact key as unique identifier enabling the link between user input of the app and data within the MC:
		`MarketingCloudSDK.sharedInstance().sfmc_setContactKey(key)`

⇒ accessToken indicates that app is granted access to the MC and to all classes and functions of the SDK	*⇒ deviceToken indicates that app is granted access to APNs*	

Figure 24: Registering the app with the Marketing Cloud (MC) and the APNs

In order to understand, what happens in the Marketing Cloud for which users and user inputs, several cases have been tested. For the user input, there are three options: the user has entered a valid email address, the user has entered an invalid email address or the user has entered no email address. If the email address was valid, then it is possible that this email address exists in the Marketing Cloud (the user is known) or it does not exist in the Marketing Cloud (user is unknown). The possible cases and how they are handled is visualized in the following figure.

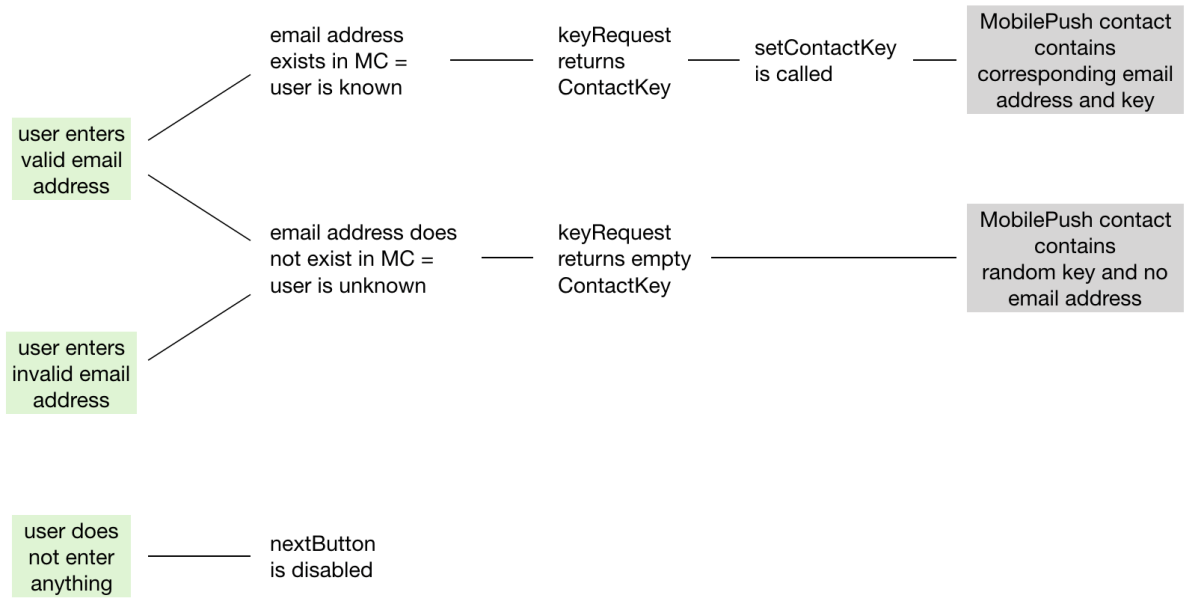


Figure 25: Possible login scenarios and what they cause in the Marketing Cloud

7 Simulation

7.1 Design

For each of the identified use cases, one concrete example push notification will be implemented. The central functionality of the ec4uApp is to receive push notifications. This was simulated by implementing one example push notification for each of the identified use cases. These use cases were determined in chapter 4.4.3. and their underlying context is the management of an event. The ones that were simulated are shown in the following table.

	Pre-Event		**Event**		**Post-Event**
	Scenario 2		Scenario 3		Scenario 4
use case	`Registration Confirmation`	`Reminder`	`Welcome`	`Update`	`Goodbye`
PN type	Inbox	time sensitive	location sensitive	location or time sensitive	location sensitive

Figure 26: Simulated use cases

7.2 Objectives

As described in the very beginning, the motivation behind push notifications is

- to replace or support existing marketing channels, which might suffer from decreasing efficiency
- to offer customers news ways of interaction to keep their loyalty or to attract new customers and
- to keep up with the most recent developments in technology and the accompanying increased expectations of customers.

More concretely, this thesis was aimed at investigating use cases for push notifications. The **research question** was:

Which use cases (with an added value) for push notifications via an app as communication channel in digital marketing can be identified?

The following simulation is meant to verify that what has been identified as potential use cases in the theory part, is feasible in reality.

7.3 Results

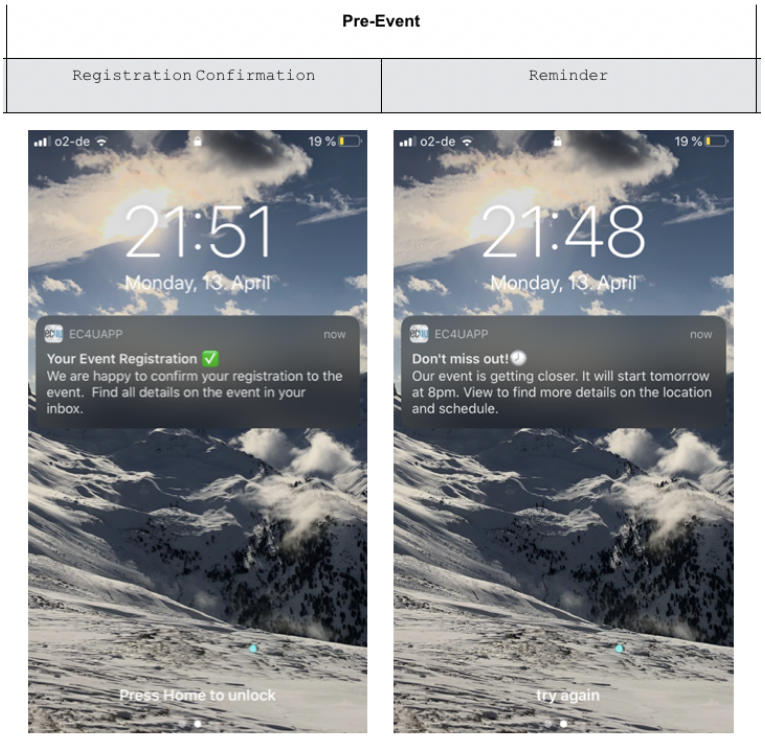
In the context of the management of an event, five different push notifications were simulated according the identified use cases. In the first case, the customer has registered for an event and gets the registration confirmation as push notification. It is sent as inbox message so that it is saved and displayed in the app. After opening the inbox push notification, the customer gets to the inbox of the app. Clicking the message in the inbox directs the customer to more information on the registration and the event. If it was not simulated, the inbox push notification would be triggered by the customer registering for an event via email.

The second use case is also part of the Pre-Event phase, but occurs directly before the event happens. Since there might pass some time between registration and the event, the customer could have forgotten about it. Therefore, shortly before the event takes place, a push notification reminds the customer of it and provides all important information, like the time, location and schedule of the event. In real life, this push notification is triggered by a set time according to the date the event takes place.

The third use case refers to the time during which the event is happening. When the customer is about to arrive at the event, a push notification is triggered welcoming the customer to the event. After opening the push notification, the customer's qr-code is displayed, which can be scanned at the entrance of the event. However, this functionality was not implemented. If not simulated, this push notification is triggered by the customer's location.

While the event is happening, another use case is to send an update on changes in the schedule via a push notification. In real-life, this push notification is triggered after the event organizer updated the schedule.

Finally, the last use case includes a push notification that the customer when leaving the event. It thanks the customer for joining and suggests to check upcoming events. In real life, this push notification is triggered by the customer's location.



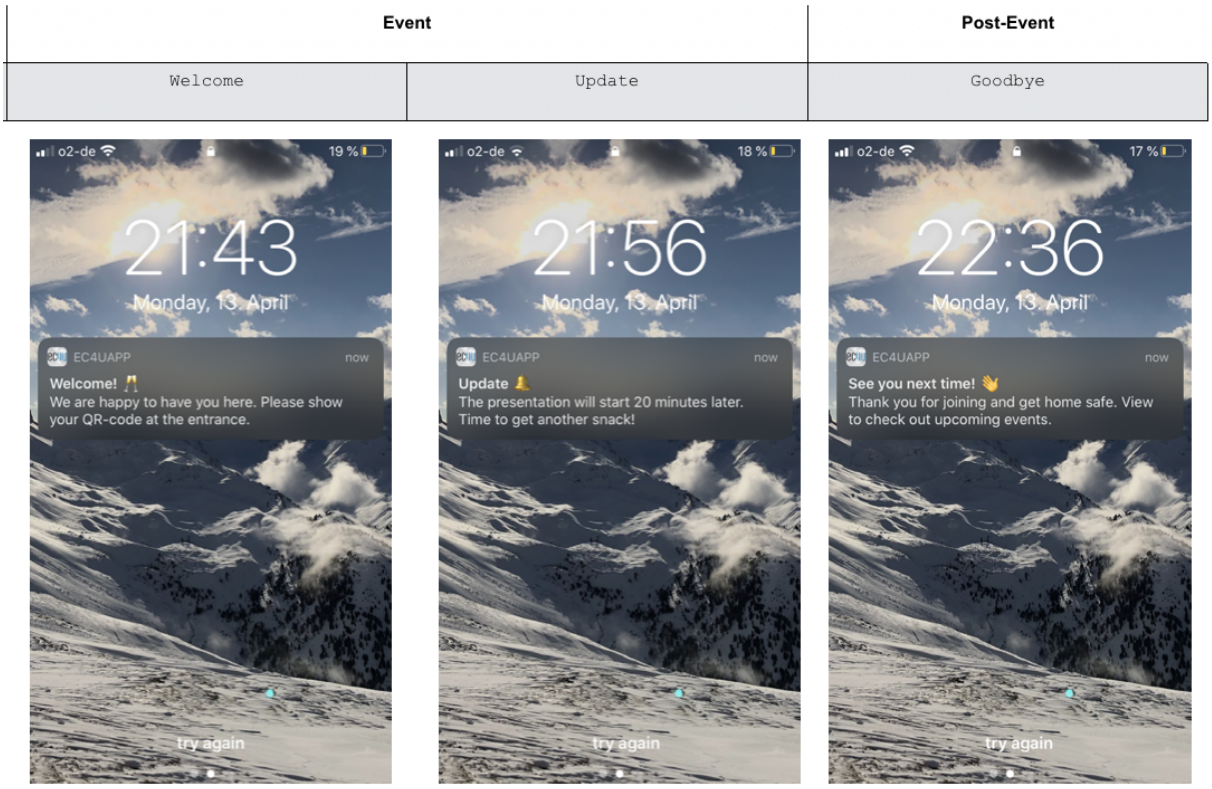


Figure 27: Simulated use cases

7.4 Findings

With these simulations, the use cases for push notifications that have been identified theoretically were implemented in reality. The simulations demonstrate concrete examples of how push notifications can be used as communication channel. This way they verify, that the identified use cases for push notifications are feasible in reality. The simulations also point out the added value through push notifications because in the implemented cases, features of push notifications are used that emails do not have. The event reminder is an example of time-critical content, which can be delivered by push notifications particularly well, as explained in chapter 2.2.3, because of their way of presenting content compact and catchy. Similarly, for sending the customer a short-term notice, like the update on changes in the event schedule, a push notification is more suitable than an email. The welcome and goodbye push notifications are cases, in which push notifications are clearly beneficial, since they are using the customer's location to trigger the notification, which would not be possible using an email.

The welcome and goodbye push notifications also demonstrate the high level of personalization. Not only is their content adapted to the customer, but also the time their are sent. This guarantees that the push notification is relevant to the customer. Therefore, the simulated push notifications meet the previously defined condition, that push notifications should exclusively be used for personal and relevant content. This condition was set to avoid that push notifications are miss- or over-used since this would make them lose their effectiveness.

As stated in the research question, the goal was to identify use cases for push notifications in digital marketing. The simulations demonstrate that the identified use cases for push notifications are feasible in reality. Furthermore, in the simulated cases, push notifications are used in a way, which would not be

possible using emails. This lets suggest that push notifications bring an added value.

7.5 Discussion

Push notifications do not only bring benefits, but also come with challenges, such as how to convince the customer to allow push notifications and prior to that, to install the app. The latter can be tackled by combining push notifications with emails. In the context of managing an event for example, the customer gets an email that includes the invitation to an event and also a link to download the app. Combining the email channel with push notifications is a good idea in general because multi-channel strategies enable more various forms of interactions and therefore more ways of adapting to the customer. Furthermore, the best channel choice depends on many factors, such as the phase in the customer lifecycle. For example, to first get in contact with the customer, their email address is required, while for frequent contact, push notifications can be more suitable. The combination of email and push notifications also makes sure that push notifications are not over-used. To conclude, push notifications can also create challenges, which could be tackled by combining them with the email channel. This, however, presupposes that contact with the customer has already been made and the email address of the customer is known.

7.6 Limitations

The simulated use cases are very specific as they are designed particularly for the context of event management. Furthermore, with the assumption that the customer's email address is known they are only applicable in later phases in the customer lifecycle, in which a relationship between company and customer already exists. For a broader use of the app, less restricted use cases have to be considered, which refer to different kinds of contexts and all phases of the customer lifecycle.

A limiting fact is also that the push notifications were simulated, which means they were sent manually so that the simulations differ from how the push notifications would work in reality. Furthermore, no real user-testing has been executed, which is why no measurable effects of push notifications are available. To be able to measure the effectiveness of push notifications compared to emails, a long-term test including various participants would be required which exceeds the scope of this project. Apart from this, not all identified use cases were implemented completely due to lack of time: the in-app push notification were left out and the qr-code of the welcome use case is missing. These are among the features that could be worked on in the future, as explained in more detail in the last chapter.

8 Summary

8.1 Status

8.1.1 Realized Goals

The overall goal of this project was to identify use cases for push notifications, which was achieved by determining them in theory and then implementing them. In order to do this, an app was developed according to specified requirements. The most important requirements were the ones related to push notifications: the app has to ask for permission to enable push notifications (FR 1), it has to be able to receive and handle them (FR 2) and finally it has to provide an inbox (FR3). The requirements were fulfilled. Another essential functionality was the login process (FR 4), which is necessary to assign the app to the right user identity. This requirement was achieved as well. Furthermore the app is filled with

content, so that the user can browse through news (FR 9) and upcoming events (FR 6). The most crucial requirement among the non-functional ones was that the app can be integrated into the existing software system, which is the Salesforce Marketing Cloud. The developed app uses the Marketing Cloud SDK and is able to receive push notifications sent by the Marketing Cloud. Therefore, this requirement was also realized. With the app, different kinds of push notifications were tested in different use cases. Out of the six identified use cases, five were implemented.

	Pre-Event			Event		Post-Event
use case	Onboarding	Registration Confirmation	Reminder	Welcome	Info/ Update	Goodbye
PN type	In-App	Inbox	time sensitive	location sensitive	location or time sensitive	location sensitive
realised?	no	yes	yes	yes	yes	yes

Figure 28: Realized use cases

8.1.2 Open Goals

As visualized in the table above, the first use case which includes in-app push notifications, was not implemented due to lack of time. Furthermore, in-app messages are not classic push notification and therefore not as relevant to this project. In the welcome use case the corresponding screen in the app displaying the qr-code, where the push notification should lead to, was not implemented. Accordingly, the requirement defining that the app can be used to check in at the event (FR 7) was not fulfilled. Furthermore, the app cannot connect to a calendar (FR 8). The profile cannot be updated (FR 5). Regarding the non-functional requirements, the utility requirements was only partly fulfilled since the app does not recognize invalid email addresses only missing user input.

8.2 Conclusion

Most important was to investigate use cases for push notifications. For this reason, the focus was put on the capability of the app to receive and handle push notifications and any functionality that was not necessary to enable it was secondary. Once the push notification capability was working, different types of push notifications were realized for various the uses cases, which proves that they are applicable as communication channel in digital marketing.

8.3 Future Work

For an extended use of push notifications as communication channel in digital marketing, a broader range of use cases would have to be considered, since this project focused on one specific context. Also issues associated with push notifications should be investigated further, such as what might motivate users to allow push notifications. Furthermore, to determine a precise added value created by push notifications, their effect would have to be measured by units like the click-through rate or return-of-investment. Furthermore, the functionality of the app could be developed further beyond the purpose of receiving and handling push notifications. The app could be made more interactive, so that the user can update their profile, save favorites or export content to other apps.

9 References

8 Types of Push Notifications To Reinforce Your Business. Rubygarage.org. (2020). Retrieved from https://rubygarage.org/blog/types-of-push-notifications.

Abbott, Lawrence (1955), Quality and Competition. New York: Columbia University Press.

Brasel, S., Gips, J. (2014). Tablets, touchscreens, and touchpads: How varying touch interfaces trigger psychological ownership and endowment. Journal Of Consumer Psychology, 24(2), 226-233. https://doi.org/10.1016/j.jcps.2013.10.003

Bruegge, B., Dutoit, A. (2004). Object-oriented software engineering (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Carter, J. (2019). More people browse on mobile but buy via desktop [Blog]. https://www.smartinsights.com/ecommerce/more-people-browse-on-mobile-but-buy-via-desktop/.

Dubey, S. (2019). 11 Types of Compelling Mobile Push Notifications That Delight Users [Blog]. https://www.moengage.com/blog/11-types-of-compelling-mobile-push-notifications-that-delight-users/.

Google: ad revenue 2001-2018 | Statista. Statista. (2020). Retrieved from https://www.statista.com/statistics/266249/advertising-revenue-of-google/.

Khandelwal, A. (2020). 9 Best Push Notification Strategies For Marketing [Blog]. https://vwo.com/blog/push-notification-marketing-strategy/.

Konus, U., Verhoef, P., Neslin, S. (2008). Multichannel Shopper Segments and Their Covariates. Journal Of Retailing, 84(4), 398-413. https://doi.org/10.1016/j.jretai.2008.09.002

Lemon, K. and Verhoef, P. (2016). Understanding Customer Experience Throughout the Customer Journey. Journal of Marketing, 80(6), pp.69-96.

Li, C. and Ma, L. (2015). Operation Design of Customer Relationship Management System. 2015 8th International Symposium on Computational Intelligence and Design (ISCID).

Marketing. (2019). Retrieved 29 December 2019, from https://en.wikipedia.org/wiki/Marketing#cite_note-3

Mezzacca, M., DuDell, M. P., Wooldridge, P. (2019). State of Online Shopping Credit Card Usage Habits In 2019. CouponFollow.

Mezzacca, M., DuDell, M., Wooldridge, P. (2019). The Millennial Shopping Report. CouponFollow.

O'Connell, C. (2015). 2015: The Year that Push Notifications Grew Up [Blog]. https://info.localytics.com/blog/2015-the-year-that-push-notifications-grew-up.

O'Connell, C. (2016). 2016: The Inside View: How Consumers Really Feel About Push Notifications [Blog]. https://info.localytics.com/blog/the-inside-view-how-consumers-really-feel-about-push-notifications.

Rapp, A., Baker, T., Bachrach, D., Ogilvie, J., Beitelspacher, L. (2015). Perceived customer showrooming behavior and the effect on retail salesperson self-efficacy and performance. Journal Of Retailing, 91(2), 358-369. https://doi.org/10.1016/j.jretai.2014.12.007

Rosenbloom, B. (2013). Marketing channels. South-Western, CENGAGE Learning.

Simoes, M. (2013). Most Small Businesses Shouldn't Even Bother With Apps [Blog]. https://www.businessinsider.in/Most-Small-Businesses-Shouldnt-Even-Bother-With-Apps/articleshow/21242253.cms.

Statista Research Department. (2020). Online purchase penetration of product and services in France as of January 2019, by type. Retrieved from https://www.statista.com/statistics/382990/leading-products-and-services-bought-online-france/

Sweezey, M. (2014). Marketing Automation For Dummies. Wiley.

Tardi, C. (2019). What Is a Value Chain?. Retrieved from https://www.investopedia.com/terms/v/valuechain.asp [Accessed 29 Dec. 2019].

The Essential Guide to the Customer Journey and Lifecycle | Customer Success Software | Gainsight. Customer Success Software | Gainsight. (2020). Retrieved from https://www.gainsight.com/guides/the-essential-guide-to-the-customer-journey-and-lifecycle/.

The Radicati Group, Inc. (2015). Email Statistics Report, 2015-2019. PALO ALTO, CA, USA. Retrieved from https://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf

Twin, A. (2019). Everything Marketing Entails. Retrieved from https://www.investopedia.com/terms/m/marketing.asp[Accessed29Dec.2019].
Watson, L. (2015). Humans have shorter attention span than goldfish thanks to smartphones. The Telegraph.

What are Push Notifications? A Complete Guide [2020]. Website. (2020). Retrieved from https://vwo.com/push-notifications/.

Zantal-Wiener, A. (2019). 7 Types of Push Notifications Users Actually Enjoy [Blog]. https://blog.hubspot.com/marketing/push-notification-types.

10 Appendix

Appendix A
University App Problem Statement

University App Problem Statement

1. The Problem

Currently, there are no applications dedicated to students in universities. Students want to participate in university courses and see announcements in these courses, but have this information spread out on multiple websites, which are hard to find. They want to find fellow students with the same interests and share their opinion about courses and course material, however this is not possible yet. They also want to discuss exam questions and find the right place where the exam takes place without searching on Google.

2. Scenarios

Arjun, an incoming student from India, is studying computer science at TUM. He has business administration as minor subject and is already used to visit the courses in the computer science building. The business administration courses however, are located in a lecture hall in another building in the city center of Munich. He never visited the other building before, so he does not know how to find the lecture halls for his minor subject. He browses through the courses in the course catalog of the University App and finds the course "Foundations of Business Administration" with course times and the location of the lecture hall on a map.

While he is attending the course, he makes contact with fellow students who also attend the course and reads their comments. He likes one comment "Great exercises" by Jenny, who is also studying informatics. From Jenny's picture, he remembers that they met a week ago at the coffee machine. He requests friendship with Jenny (she might help him to pass the final) and adds a new comment about exam questions from earlier exams. While he is browsing, Jenny is notified about the friend request and accepts it. Arjun, in turn, is notified that Jenny has accepted his request and now browses through all the courses that Jenny is visiting. He finds another interesting course "Cost Accounting" that he wants to visit and saves it into his course list.

3. Requirements

The following functional requirements (FR) and nonfunctional requirements (NFR) have to be addressed in the project.

FR1: Search for available courses: A student can see all courses of the current semester in his major and minor subject. He is able to join the course which saves it into his course list. He can also drop a course.

FR2: Check course details: A student can see details about a course such as the course times, the location of the lecture hall on a map and other course attendees including their name and picture.

FR3: Update profile: A student can update his profile settings and his profile picture. He can also change the notification settings.

FR4: Add comments: A student can add comments about a course and thus start a discussion. Others can like the comment and write follow-up comments.

FR5: Request friendship: A student can request friendship with another student who then receives a notification about the request. The second student can accept and reject friendship which both notifies the first student.

FR6: Browse friends' courses: A student can browse the courses of his friends.

FR7: View announcements: A student can view course announcements and comment/like them.

FR8: Post updates to timeline: A student can post updates to his timeline. Friends are notified about updates and can comment and like them. Certain updates are posted automatically such as saving a course into the course list or commenting on a course.

FR9: See course calendar: A student can see all courses in a calendar.

NFR1: Usability: The app should be intuitive to use and the user interface should be easy to understand. All interactions should be completed in less than three clicks.

NFR2: Conformance to guidelines: The design of the app should conform to the usability guidelines for the chosen operating system.

NFR3: Target platform: The app has to be developed in Java.

NFR4: Backend system: The customer provides a backend system with a couple of services that have to be used in the app.

Additional constraints:

- The version control system must be git.
- Source code documentation must be in HTML format.

4. Target Environment

The application should be demonstrated in Java.

5. Deliverables

- Requirements Analysis Document (RAD)
- System Design Document (SDD)
- Source code under version control including source code documentation

6. Client Acceptance Criteria

The app must demonstrate at least the following functionality: It shows a list of courses that a student can join. The student can see the attendees of a course with their picture and can request friendship with other students. The app communicates with the backend system provided by the customer and conforms to the usability requirements.

Appendix B

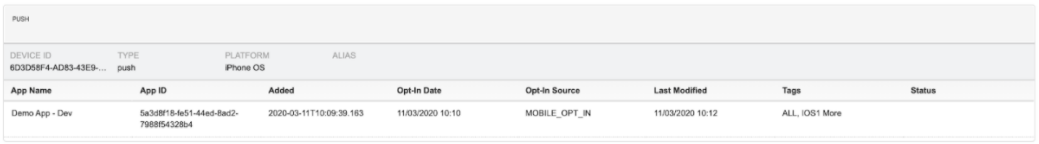
JSON object returned from key request

<table>
<tr><th>key</th><th>value</th></tr>
<tr><td>"channelAddressResponseEntities":</td><td>
<table>
<tr><th>key</th><th>value</th></tr>
<tr><td>[{"contactKeyDetails":</td><td>[{"contactKey":"0031r00002lqJirAAE",
"createDate":"2020-02-10T09:35:00"}],</td></tr>
<tr><td>"channelAddress"</td><td>"katharinameitzler.ec4u+1002@gmail.com"}],</td></tr>
</table>
</td></tr>
<tr><td>"requestServiceMessageID":</td><td>"b03612b3-0565-43b8-94ac-30d09004da23",</td></tr>
<tr><td>"responseDateTime":</td><td>"2020-03-10T08:56:13.3187284-06:00",</td></tr>
<tr><td>"resultMessages":</td><td>[],</td></tr>
<tr><td>"serviceMessageID":</td><td>51f162a3-9f51-47b8-a927-f48673af271a"}</td></tr>
</table>

Appendix C

Contacts in MobilePush created according to user input

Contact mit random key, für optin ja:

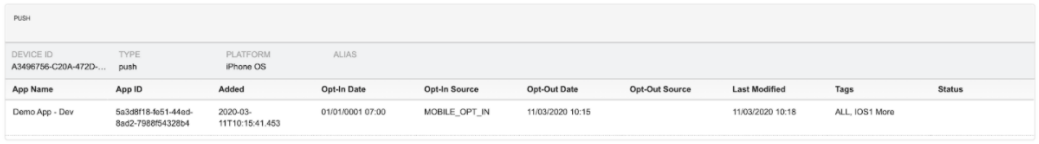


PUSH

DEVICE ID	TYPE	PLATFORM	ALIAS
6D3D58F4-AD83-43E9-...	push	iPhone OS	

App Name	App ID	Added	Opt-In Date	Opt-In Source	Last Modified	Tags	Status
Demo App - Dev	5a3d8f18-fe51-44ed-8ad2-7988f54328b4	2020-03-11T10:09:39.163	11/03/2020 10:10	MOBILE_OPT_IN	11/03/2020 10:12	ALL, iOS1 More	

Contact mit random key, für optin nein:



PUSH

DEVICE ID	TYPE	PLATFORM	ALIAS
A3496756-C20A-472D-...	push	iPhone OS	

App Name	App ID	Added	Opt-In Date	Opt-In Source	Opt-Out Date	Opt-Out Source	Last Modified	Tags	Status
Demo App - Dev	5a3d8f18-fe51-44ed-8ad2-7988f54328b4	2020-03-11T10:15:41.453	01/01/0001 07:00	MOBILE_OPT_IN	11/03/2020 10:15		11/03/2020 10:18	ALL, iOS1 More	