Project title: “MembaShip”

1. Introduction

Problem Statement:

The climbing walls I use in Bristol each come with their own membership card that has a unique barcode printed on it that is scanned every time you climb. Each card takes up the limited space in a wallet or purse which could mean that curtain cards wouldn’t be kept in there until they are needed. This creates the problem of forgetting your cards and leaving them behind when you need them.

Our phones hold a high priority in our lives, so we take better care not to lose it and keep it on us, it’s rare for us to not have our phone nearby. So I’m going to create an app that stores all your membership card details (barcodes that you scan on entry) within it to save the user having to carry every card they use on their person and eliminating the potential struggle of rummaging through their wallet or purse to find the card they need.

1. Method

Suitable Tools:

With the advice of my supervisor I will be designing my app with JavaScript along with its libraries, mostly react.js. I could end up using AJAX, if this is the case then ill also utilise PHP to handle the data that is returned and to write the pages/files that AJAX is calling too. Then use visual studio code as my IDE.

It makes sense for me to use these coding technologies as these are the ones I’ve had the most experience with excluding react.js, I’ll need to do some research into this technology.

There is also the option for me to implement a framework like Ionic and React Native, but I feel ill gain a better learning experience building my app from the ground up on my own.

For the UML I’ll be using astah for the same reasons for my code language choices, as this is the software I have the most experience in for producing UML.

Methodology:

I have chosen extreme programming as the methodology ill be using due to its focus on frequent releases in short development sprints that encourage change when needed. Therefore, I can retroactively reflect on my work and make changes before moving onto the next section of the project.

Extreme programming also requires developers to plan and understand the customers user stories. This is also a benefit for me as I will start coding after I have created the user stories and UML.

The practice of designing with simplicity in mind and testing your code often are practices that I believe will be a successful routine for me and the project.

You can read the detail in the article (LeanKit, 2019).

Risks:

The organisations issuing the membership cards taking issue with the app therefore not being happy with the card details being used within a third-party setup.

Another program/app existing that works either similarly or completely like my proposed idea would.

Schedule/deadline issues with my other modules affecting the time I’m dedicating to this project.

Issues with learning the new and already known languages I need to use.

Having not used UML and use cases for a while this could pose some time-wasting issues.

Version Control:

<https://gitlab.uwe.ac.uk/c23-day/dissertation>

1. Research

Primary Research:

When I was asking the climbing walls around Bristol for their permission to use the membership cards they supply, one of the companies highlighted the existence of an existing app that provides the same functionality that I’m aiming to provide called “Stocard”. After finding “Stocard” I was suggested more apps that work the same from the play store. I think this would be a good reference point for the design of my app as I can highlight what these apps do well and the areas I can improve or create if it’s lacking some functionality.

Stocard Pros:

* UI is simple and easy to understand and use
* The app provides exclusive offers and coupons to its users
* Can store bank cards
* Uses fingerprint scans and code security, can freeze cards
* The app works well on smartwatches
* Well reviewed, 4.8\*/ 5\* out of 500k reviews, 10M+ downloads, implies very few bugs

Stocard Cons:

* Although the UI is simple, It’s a little bland
* Finite number of free spots to use bank cards, pay needed otherwise

Cards Pro:

* Prettier UI
* Has the option to open any card instantly from outside the app
* Offers same functionality and security as Stocard
* Well reviewed, 4.6\*/5\* out of 61k reviews, 1M+ downloads

VirtualCards Pros:

* Can create an account to act as a card backup
* Can apply for new cards from select partners
* Contains a shopping list you can populate with desired items and any offers relating to those items are displayed
* Users can use their voice to add items to the shopping list
* Well reviewed, 4.7\*/5\* out of 21k reviews, 500k+ downloads

VirtualCards Cons:

* Similar UI to Stocard
* Can’t be used on smartwatches
* Can’t be used with bank cards

From this research I’ve gained a solid perspective on the market for these types of apps. With combined downloads of 11M+ there is definitely a demand for this type of service. It also seems users are more interested in the app’s functionality over its presentation with these types of services, I feel this is true as the reviews are exceptionally high for the apps which have bland UI experiences. This gives me a lot of freedom with the UI as the example set from these apps is relatively low.

Along with some ideas to investigate.

Smartwatches

Bank card option

<https://www.statista.com/statistics/515634/wearables-shipments-worldwide-by-vendor/>

Informal Sources (User Stories):

Functional/Non-Functional Requirements:

Previous Case-studies:

Technology Selection (Pugh Matrix):

References:

LeanKit (2019) *learning and Education* Available from: <https://leankit.com/blog/2019/03/top-6-software-development-methodologies/> [Accessed 02/02/2020].