Computer Science Program

3rd Semester 2015

Semester Project

Implementation of the POCA app

The report has been prepared by :

Alexandru Draghi

Ionut Danci Bumbea

Leonard Hirja

Richard Retezi

Supervisor:

Gianna Bellé

Date of submission:

14-12-2015

Real Life Education

Table of Content

1. Introduction

"There are no problems, only challenges."

Selecting a project that fits the requirements is not hard. There are tons of examples and samples on the internet for many kind of tasks. Selecting a project that is somewhat original, fresh, interesting and challenging is another story. Group 2 decided to go for the latter.

The main requirements for this semester were to code in C#, to have a client/server connection, high complexity, concurrency, and to use some kind of security measures. Regarding to the system development and design, Group 2 had to come up with an idea that can satisfy the above mentioned criteria, create user stories and story cards, design the architecture and interface of the software, plan the whole project, decide on the methodology, the group work and on the schedule.

In the following part the birth of the core idea – inception- is being detailed, as well as how Group 2 managed to develop a project from that single idea.

2. Idea Generation

At the start an idea for the project was needed. The members of Group 2 started to write down ideas on different sheets of papers. After everyone had written down at least 4-5 ideas individually, these ideas were evaluated.

There was one single rule: There are no rules. Group 2 let its imagination fly.

Some ideas were very distant from each other, while others were relatively similar. The ideas were grouped together according to similar thematic and/or platform. After the grouping, each idea was anonymously evaluated. Some of them were discarded instantly, some were kept for future possible consideration or for reserve, while some made the best candidates for the project. A short list was created and in the end, one idea was selected for the project as a common agreement.

In the next section, a short description of the selected idea is presented.

2.1 POCA Mobile Application

The idea was simple: There are lots of young professionals, scattered around the world, who are really interested and/or good in a given area of study and there is a matter they would like to discuss with people with the similar interests. Using modern technology, it would be really nice if they had an app that can find people who share the same passion and provide a field for them to be able to contact and communicate to each other. Using a mobile application they could search through different predefined "passions" and find people of the same interest within their area of living or even anywhere else around the world. When a user finds a person with the same interest he/she can send a request for contact and that person can accept the request. Then, a match is being established and they can start discussing the topics they'd like via chat. In case of multiple people want to discuss a matter, a group chat is possible. This idea can help lots of people who need a platform to share their thoughts and ideas with others who could be interested in the topic.

In the following section an example of a User Stories is shown as example.

2.2 User story[1]

Persona:

Name: Martin Cartridge

Age: 25

Location: Bristol, UK

Occupation: Entrepreneur

Motto: "You cannot know if it's a failure until you've tried"

Story card:

AS A young professional I WANT TO find others with the same area of interests SO I CAN discuss establishing a start-up company with them.

Scenario:

Martin has recently graduated from University and started working in a multinational company. He's not satisfied with his current position. He's thinking about launching a start-up company. He has heard about this new app – POCA – from a friend of his. On his way back on the train, he decides to give it a try. He quickly registers by filling some personal information (name, age, location, etc.). He selects 3 of the predefined passions: Entrepreneurship, Economics and IT. He then enters the main page. Martin then searches for people who are also interested in Entrepreneurship. Martin would like to see how many people are interested in this area close to his city, Bristol. He can only find a limited amount of persons – 2 – in his city. Martin then tries to widen the search to see how many people are interested in this Passion in England. He can find significantly more people – 45 --, to connect with. He also tries out a worldwide search to check possible future connections from other countries and he finds – 1700 – users. Martin then chooses 4 persons he could sympathize with and sends a connection request. Martin then waits for his request to be accepted by the other users. After Martin has arrived home, he checks POCA again and sees that 2 users have accepted his request. He writes them a short message, using chat mode. He manages to start a conversation with one of them about how to start a small business.

3. From Idea to Materialization

Vision Statement

FOR people who share the same passion, WHO NEED to connect with like-minded thinkers, POCA IS A mobile app THAT provides the ultimate platform for connecting young professional thinkers. UNLIKE forums, social networks and mainstream communicating platforms, POCA OFFERS an easy and fast way to search and find people within the same area of study and communicate with them.

In this section the process of creating a real-life project from an idea is presented. Now that Group 2 has defined the idea, planning could be started. The first order of business was to check the requirements for the project and align it with the idea.

3.1 Meeting the requirements

The Functional requirements for this project are to use C# programming language, to have a high complexity program, which is able to handle Concurrency, have a database connection and most importantly, to have a client-server connection. The project also has to focus on technology requirements (IT Security measures, protocols, architecture, etc.), and to have a web client that connects to a web server. Most projects are using a website to make the connection, but our idea indicates a mobile application. For the reason that we have to use C#, our platform has to be a Windows Phone that runs on Windows 10 Operating System, which is connected to the WCF (Windows Communications Framework). We need to have a web server connection, therefore we also have to make a sample website for our idea and have a two-sided connection. A Dedicated Client – the mobile app – and the Web Client (sample website). More about the architecture can be read in the Technology Report.

Group 2 also tried to pay attention to Non-Functional Requirements in which the most important features are Usability, Performance, Portability and Security. The characteristic of the project that it is primarily a phone application requires that it is easy for the users to handle, its fast, it can run on multiple phone devices – although because of the requirements (C#), it will run only on windows phone firs – and it has high security to protect the private data of the users and defend against viruses and spam.

3.2 Agile Development[2] and XP[3]

At the start of the project Group 2 had to decide what Development methodology to take: Waterfall or Agile. The project indicates multiple uncertainties, and the group lacks of several key information at the start of the project, therefore a dynamic, flexible approach is needed in order to carefully plan the program. Agile Development is a great way to that, because it lets the members of the group to give a quick, efficient response to unknown occurrences.

Group 2 will use XP (eXtreme Programming), by implementing the incremental delivery approach. This helps the team to break down the projects into several parts, priorities, and ensure that the top priority functions of the program will be shipped and provides possibility for a dynamic response to changes. Some of the functions of our project are significantly more complex and challenging than other – still very important, but less harder to program – functionalities, therefore dividing the priorities into separate iterations/increments proved to be an important task. The most complex function was sure to be coding the search algorithm, there for Group 2 decided to take it over two increments with less complex functionalities (registration, log in).

XP practices were implemented in the project:

- Planning game: User stories and story points were created and evaluated accordingly.

- Small releases: The first release will contain the most important core functions and latter releases will add additional new functions.
- Metaphor: Group 2 decided upon a common framework to work within.
- Simple design: The design for the software will be minimalistic and concentrates on functionality.
- Define Test First: Group 2 designed the architecture and tested the service connection before any coding.
- Refactoring: After the first release and a well-functioned program, design will be revised and improved.
- Pair Programming: PP was introduced which enhanced efficiency.
- Collective Ownership: All the team members could change the software, but only after common agreement.
- Continuous Integration: Group 2 built the program step by step, day by day, exponentially expanding the software.
- 37 hours per week: Group 2 has worked together 3-4 hours and discussed reached milestones, errors and following steps. The group members then worked separately on their own tasks in class or from home another 3-4 hours.
- On-site customer: The product owner is a member of the group, differed in each sprint.
- Coding standards: Naturally, the group implemented many principles and practices to follow and stick with.

Combining XP with Scrum led to a powerful source for planning the project, dividing the iterations into sprints and tasks. The project will consist of 4 sprints (excluding sprint 0) with the implementation of the top 5 priority functions. The different sprints are designed in a way that they equally share a certain amount of tasks which are possible to be completed within the given time period (one sprint should take 1.5 weeks).

3.3 Risk Analysis[4]

In the project Group 2 had to take risks into consideration, therefore it is possible to measure and handle possible faults beforehand and give them a quick and valid response.

Risk	Probability	Effects	Mitigation
Prolonged illness (missing out meetings)	Moderate	Serious	Work from Home
Technical issues (old laptops with possibility of breaking down)	High	Serious	Make schedule for repairing and upgrading laptop
Requirement specifications change original optimal platform selection	Moderate	Serious	Switching to optimal platform
Search algorithm proves to more complex than thought before	Moderate	Catastrophic	Change schedule to add more time code Search Algorithm
Predefined connection architecture won't work in reality	Low	Catastrophic	Restructure architecture and fix connection errors
Potential users are not satisfied with the shipped first release	Low	Catastrophic	Make survey to ask for user reviews to optimize UI and functionality

3.4 Quality Attributes[5]

Selecting and focusing on some of the many quality attributes are essential to reach a high quality user experience for the program. The main quality attributes the product focuses on are:

- Conceptual Integrity: The software is coherent, consistent in design.
- Maintainability: The program is designed in a way that it is easy to maintain, upgrade, improve or change if necessary.
- Availability: The software is ready to be used by the customers with its core features functioning.

- Interoperability: It is delayed to future implementation, primarily it runs on Windows Phone because of C#. It is planned to be able to run on Android and iOS phones.
- Manageability: It is easy for the administrators to manage the software and intervene in case of errors, viruses, spam attacks, harmful users, etc.
- Security: As probably the most important feature, security protects the users' personal data and exchange of information.
- Scalability: As the project includes the possibility that many users can register within a short amount of time, the system has to be able to handle a sudden increase of data flow (load) without setback in performance.
- Usability: It is extremely important for Group 2 that the project meets the requirements, implements the core functionalities in its first release and it is easy to use for the users which will result in positive user feedbacks.

Naturally, all the other quality attributes were taken into consideration as well, such as Reusability, Performance, Reliability, Supportability and Testability.

4. Project Implementation[6]

Group 2 drew mock-ups for the project idea, thought about the architecture and collected user stories in order to have a deeper understanding of the project. After the members collected all the possible user stories they could think of, story points were added to them, so that prioritization will be possible. A constant discussion within the group – when deciding on priorities – is that whether the most complex user story the top priority or the one which is simpler, but vital to the project. When deciding on story points, the most complex user stories received the highest score, but that doesn't necessarily mean that those became the top most priorities, but the ones that are vital.

Within this framework, the group had selected the top 5 priorities user stories and broke it down into tasks. The top 5 priorities were the following: Register and Login (1st), Search (2nd), Request and Match (3rd), Receive and Send message (4th) and Book an Advisor (5th). Even though Search received the highest score because of its complexity (19 pts in total – with subtasks), Register and Login became the top most priority (with 13 story points), because without it the application cannot exist and it is essential for registering and protecting user information. Search received the highest score because the estimated time for writing the search algorithm considered to be the most complex and challenging among rest of the user stories and it is one of the core functionalities of the app. Match and Messaging (or Chat) are also core functionalities and Book an Advisor was important to include because that provides the Concurrency (in order to align with the requirements). Originally another user story was set as 5th priority: Group Chat which allows the users to communicate with multiple other users and share their ideas. Regarding to the fact that the project had to align with the concurrency requirement, Group Chat was moved from the First Release and Book an Advisor took its place.

4.1 Sprint 0

Sprint 0 is the initialization of the project, where all the planning, architecture related decisions and design related solutions are explored. Group 2 had drawn a mock-up for the project as well as a basic architecture in order to have a better understanding of the planned software. A decision was made on which diagrams are needed to help the team in the progress.

The selected diagrams were the following: Use Case Diagram, Domain Model, Candidate Class Diagram, System Sequence Diagrams and Design Class Diagram. These diagrams are important in order to visualize connections and relations in the program.

Group meetings and stand up meetings were also declared during Sprint 0. The team decided to meet at UCN's Sofiendalsvej campus at 8.30 when there's class in the schedule, otherwise to meet at Leonard's and Alex's dormitory where there is a study room perfectly fitting for the group's needs and it is the closest place to work on the project for the most of us. The meeting time at the dormitory was set to 11.00. The group had divided the upcoming tasks, decided on the Scrum Master and on the Project Owner for each sprint, and then started to prepare for Sprint 1.

As a kickstart to the project, Group 2 created the Product Backlog which contains all the user stories, even the ones that are not in the top 5 priority. Then the team created the Sprint Task Table and prepared putting the user stories and their related tasks for Sprint 1. A scheme for Sprint Backlog was also created, so the members can the amount of remaining workhours for each task in each user story per each sprint. The values were calculated in a way, that the group decided on 8 hours / day of work in which 4 hours are for coding only, whilst the other 4 hours are for writing the report, discussing problems and also populate and modify are charts and tables (such as the sprint backlog itself). These charts were created in digital format, as it is easier for the members to input, change data and share files through GitHub independently from an exact physical location.

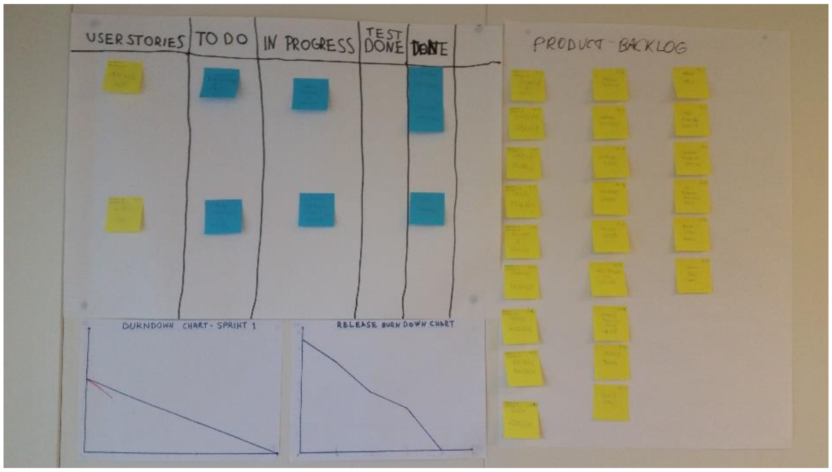
The group also created some physical charts for representation. Sprint Task Board, Release Burn Down Chart and Sprint Burn Down chart were created for that purpose. Sprint Task Board consists of the following columns: User Stories, Tasks, To Do, In Progress, Test (test phase) and Done columns. The tasks (which are on sticky notes) are moved between the columns of the board accordingly. The Sprint Burn Down Chart represents the correlation between the amount of story points and the time (number of days) spent on working per sprint. Release Burn Down chart is aggregating all of the Sprint and it shows the total sequence in one single chart.

After all the preparation, the team was ready for Sprint 1.

4.2 Sprint 1

Sprint 1 consisted of two main areas: Setting up the database and making the connections with the software via the two primary user stories: Register and Login. Register and Login were selected as the first two user stories for sprint 1 because even though they are pretty simple, they are indispensable and the program cannot start/function without them. Creating the database, setting up the empty tables and queries were relatively simple as the group also had experience with these tasks. Creating the interface for register and login was also simple and was done within one day.

As mentioned before, the target platform was Windows Phone 8.1. As the team had experience on developing small applications (games) on windows phone before and also had experience working with databases, the members thought that after a short period of research on architectures (connections between databases and windows phone), it will be possible to connect the database with our chosen platform. Unfortunately the group had faced some serious obstacles and challenges.



The team decided to make a test first in order to test the connection. The result of the test was negative, meaning that even though the separate elements of the program (the classes, the model layer, the database, the interface, etc.) were programmed well the connection itself failed. Group 2 was stuck on this problem for two and a half days. The connection was tested multiple times with slight changes and failed all over again. The members decided to step back, get a fresh air and think on the problem. The team revised the classes, the database set ups and the architecture. The team revised how connections and the settings of the connection needed to be done between the databases, the software and the target platform. The program was tested on the phone and also on an Emulator (Visual Studio's windows phone 8.1 emulator). Because the project requirements include a segment which is to create a web application (connect to a web server and to a website), the members decided to test the project if it is working on a web app.

According to the result the program was running perfectly as a web application, therefore the problem was the connection between the database and the phone. Our investigation of the process declared that there is a lack of support between MS SQL and Windows phone 8.1, therefore even though the code itself was written perfectly the connection couldn't be established. After all different approaches were tested and tried a decision was made that the web application is the best possible solution at this point in order to reach a stable well-functioning and runnable program.

The group was discussing and switching between the pros and cons of such a twist, but in order to be able to "ship" the project in time, we were willing to take this risk and proceeded with the Web application structure.

4.3 Sprint 2

In Sprint 2, the task was to create the Search algorithm; the interface for search, filtering and listing the searched users; troubleshoot of the errors made in Sprint 1 and align with the new structure.

The troubleshooting had taken the priority as the time went by, every single double-check and test had to be made in order to completely cover the possibilities of running the program on Windows Phone and also to be able to move to the Web Application structure. After all area has been covered, we were settled with the web application structure.

The team also had to sort out some communication difficulties that surfaced in Sprint 1. Some of the division between the tasks were unclear, so as the milestones and task durations that were previously set were in delay. Some of the team members had also a slight attendance and communication problem. Leonard had to leave couple of times due to medical issues and also attend to some examinations outside of Denmark. Richard has received a good company work related possibility for his specialization and internship, so soon he also had to take his departure from Denmark. A discussion had been started in order to solve communication problems and to find a way in order to overcome the challenges and finish the project as planned. Thankfully modern IT communication tools are existing precisely for this matter. Using chat-based online tools – such as Facebook Messenger – and occasionally with the help of Skype conference-calls were possible to enhance communication.

After all the tests and double-checks the new system architecture – with the web application – was set up, parallel to the windows phone project which was still kept in case we had the time and possibility to solve it and implement it in our project. After connecting everything in the new structure (connecting our website to the webserver, to the WCF and to our Database… detailed in the technology report), we could finally focus on the programming of the rest of the functions.

Unfortunately troubleshooting summed up a lot of time from Sprint 2 so we had to make up some time in Sprint 3.

4.4 Sprint 3

Ionut and Alex had started working on the Search function while Richard was helping out and was writing the report. Leonard was also helping out with solving upcoming challenges and function-related problems.

Alex had created the Search interface and started investigating the different search algorithm possibilities in order program this function. Ionut also joined in the writing of the algorithm.

In the meantime, Ionut and Leonard were revising the architecture and server connections once again, also investigating the possibilities of using VPN and Wireless in order to connect to the network, so as LAN Wireless in order to connect to the program using internal wireless network. Our program was hosted on one of Microsoft's newer features, Microsoft Azure. Azure is a cloud-based system that is especially designed for programmers to host and test programs created in Visual Studio (also a Microsoft product). Azure helped us to test our program how it runs in a secure cloud environment.

Given the nature of this rescheduling, modifications in our original working plan were necessary to be made. As the original requirements for our project states that for our first release we need to have a running, well-functioning program, that connects to a server and uses databases; runs on a Web Application and was coded in C# and consists concurrency, all efforts were reallocated in order meet this aims until deadline. That means that for our first release some of the less necessary and important functions were delegated into a later release state or to the "About to be implemented" section. Thankfully the Agile Development Method helped us to handle this huge change. If we had used the Waterfall development method or classic Scrum, we wouldn't have been able to adapt to such sudden and great challenges and twists in our project. Of course, because waterfall development includes that we have all required information, input and planned output designed for the project, we had to start with Agile. Agile included that we lacked some important information and it included the possibility for error handling and to be able to implement Change Management.

The coding part of the following functions were relatively easy compared to the huge challenges we have faced before. Search was created successfully and we could proceed on to the next Sprint.

4.4 Sprint 4

Sprint 4 was to implement the concurrency and finalize the program. Concurrency in this program meant a so-called "Premium Service" option were users can subscribe to and to be able to use some advanced features of the program such as booking an advisor. In relation to the service our POCA app offers is that the users who share the same "Passion" can meet and share their ideas, thoughts and plans and maybe to initiate a cooperation among themselves. Booking and advisor is there to help in case e.g.: the Passion the users share is "entrepreneurship" or "launching a start-up company". Maybe the users have great ideas, but they don't know how to manifest them, how to find the right people needed in order to materialize this idea, etc. With booking an advisor and scheduling an online meeting the given advisor can help with finding the right people: For example the user has an idea of a business and knows programming, but the user doesn't know how to launch a company and how many and what kind of professional people he or she needs in order to make this work. After subscribing to the premium service and scheduling a meeting with an advisor, the advisor who is professional at this field, can help the user out and tell him that he needs a Financial Analyst, an Economist, a marketing professional, etc. The Advisor can also suggest and share ideas how the procedure of launching a company works and what other things are needed to be done in order to achieve it. The core function of this user story is to pick an advisor and an appointment with her, select time for a scheduled meeting and book it. Concurrency is included when two or more users try to book the same meeting time at the same time.

The team had experience of handling concurrency from the previous project where a "Book Table" function was introduced to our restaurant software. It was about reordering and booking tables for customers. In the POCA app, booking an appointment with an advisor is not far from this function in implementation terms. With the usage of the past experience coding the concurrency was easier to manage and the team could manage to move on to finalize the program.

In the finalization all classes, layers were checked, the program was tested and ran multiple times and architecture was revised over and over in order to reach the 100% result and the meeting with the requirements. Although we tried to test and make the Windows Phone platform work as well, but with the lack of time and resources it was delayed to a later release of the software. The program is working just fine using a web application and it is stable, therefore the group can be delighted after all that challenges we had to face with.

5. Conclusion

The POCA app was the best idea among many the members of the group could come up with. It consists a huge potential and later releases and fixes can add true value. The group started working on this project with great enthusiasm and positiveness. We decided to take on a big challenge and implement our program on a Phone. Unfortunately the challenge was far greater than we thought before and it was not because of the lack of knowledge or skills, but because of the lack of software support among the different IT tools. We have spent a lot of time trying to solve the challenge, but in the end we switched in order to meet with the original requirements and launch our app on a web, using a website and a webserver. The rest of the designing and a programming went well how it was supposed to be.

The group believes that the fact and the volume of the challenge and obstacles we have faced with and that we could take the responsibility and the risk to switch to the web application means that the Agile Development Method worked and it was immensely crucial and useful for us.

Although we had some communication errors and time-lags we could resolve it in the end and work together in order to finalize the project. This project has shown us what huge risks and challenge can surface when choosing a new architecture or platform, how the lack of support between software can ruin everything, how important internal communication is and how to tackle down obstacles even if it means to take responsibility and risks in order to meet with the requirements.

We have learned what our individual and group-wise mistakes are and how to resolve or minimize them for the sake of the success of the project. This project had a big and hard learning curve and we consider it as a huge personal and professional development for our team.

6. References

1 – Suzanne Ginsburg: Prototyping App Concepts, page 79-86 (Analyzing User Research) & page 171 (Usability testing app concepts)

2- Ian Sommerville – Software Engineering (2015), Chapter 2

3 – Kent Beck – Extreme Programming Explained (1999) – Chapter 2-6

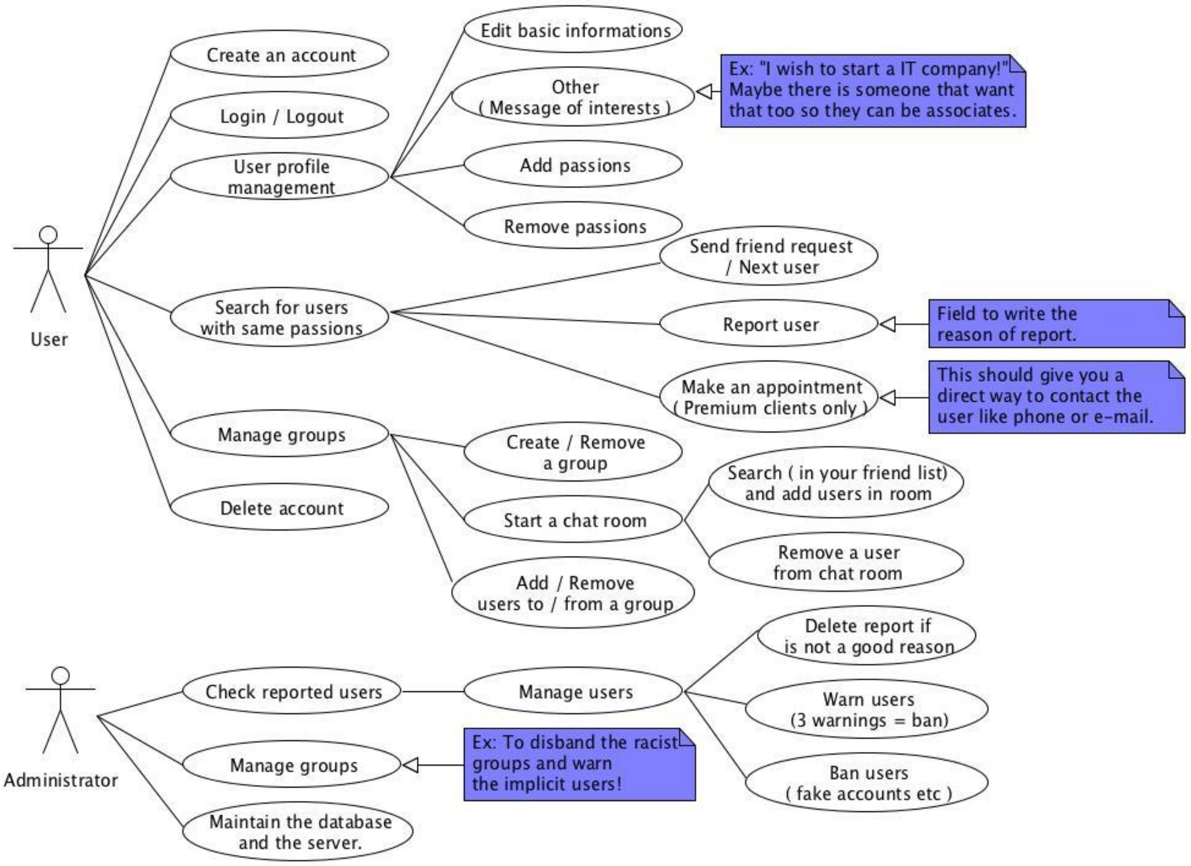
4- Kent Beck – Extreme Programming Explained (1999) – Chapter 1

5- Ian Sommerville – Software Engineering (2015), Chapter 22-25

6- Kniberg Scrum and Xp from the Trenches (2015), Chapter 3-5

7. Appendices

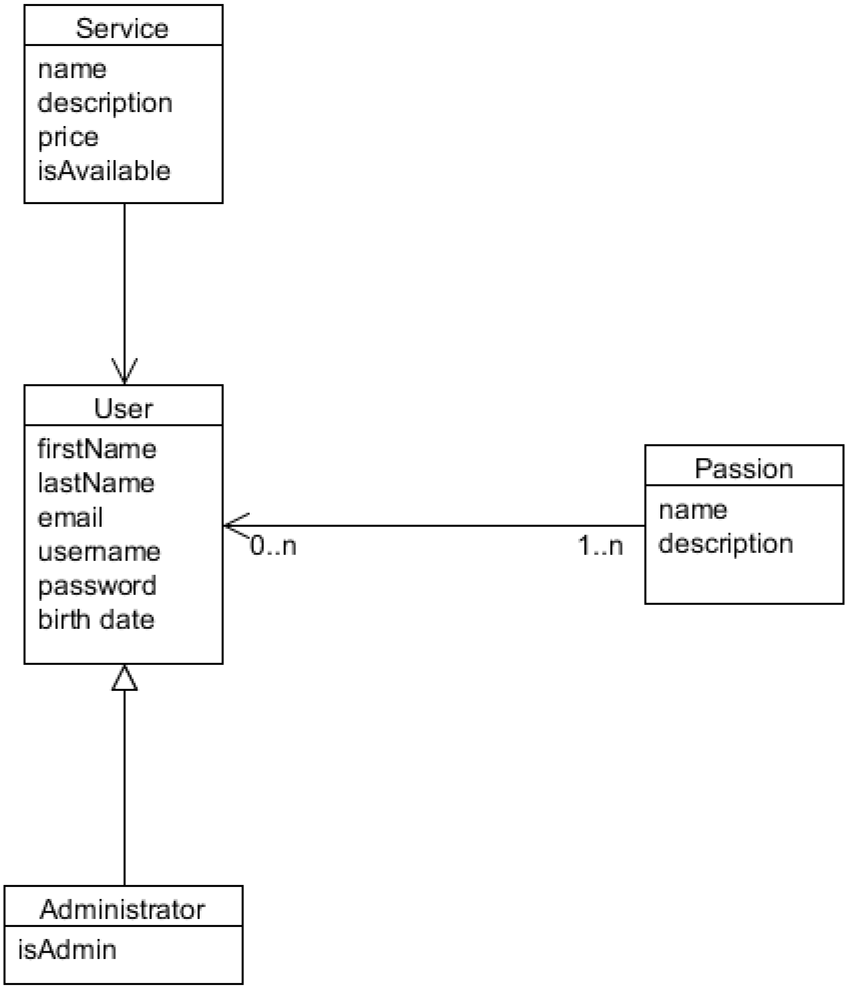
User Case Diagram:


User
Create an account
Login / Logout
User profile management
Edit basic informations
Other (Message of interests)
Ex: "I wish to start a IT company!" Maybe there is someone that want that too so they can be associates.
Add passions
Remove passions
Search for users with same passions
Send friend request / Next user
Report user
Field to write the reason of report.
Make an appointment (Premium clients only)
This should give you a direct way to contact the user like phone or e-mail.
Manage groups
Create / Remove a group
Start a chat room
Search (in your friend list) and add users in room
Remove a user from chat room
Add / Remove users to / from a group
Delete account
Administrator
Check reported users
Manage users
Delete report if is not a good reason
Warn users (3 warnings = ban)
Ban users (fake accounts etc)
Manage groups
Ex: To disband the racist groups and warn the implicit users!
Maintain the database and the server.

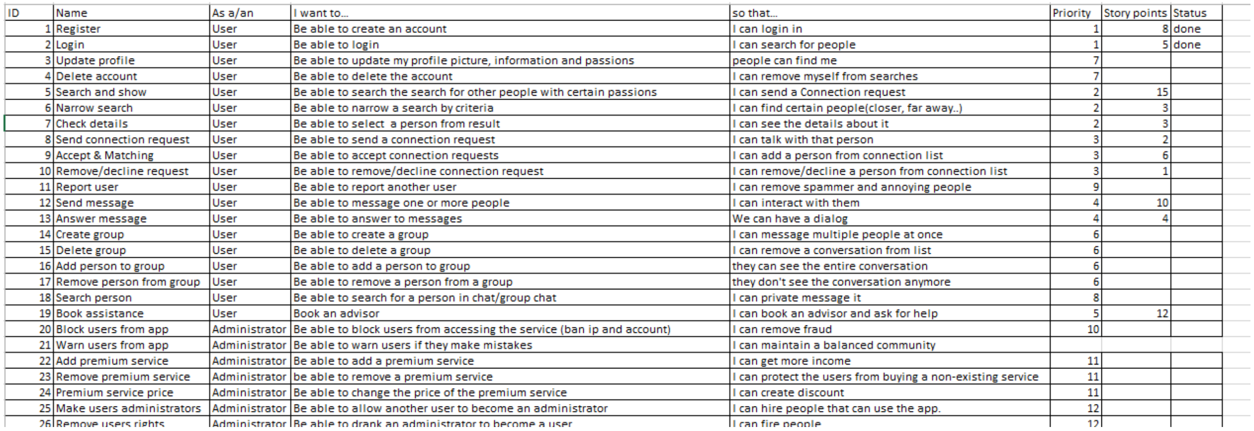
Candidate Classes:

Candidate Classes
User
Premium User
Admin
Advisor
Reserve
Message
Search
Matching

Domain Model:

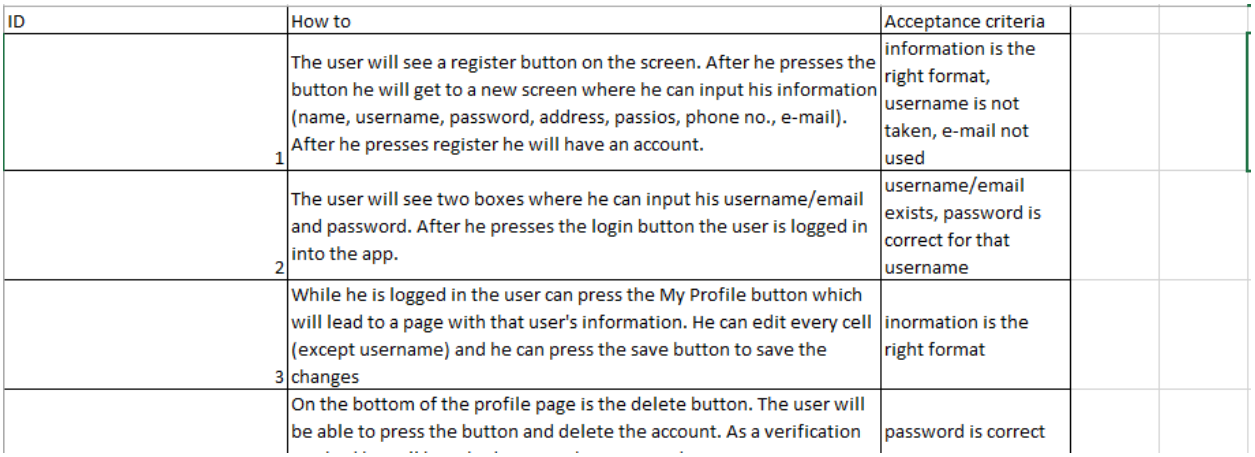


Product Backlog:



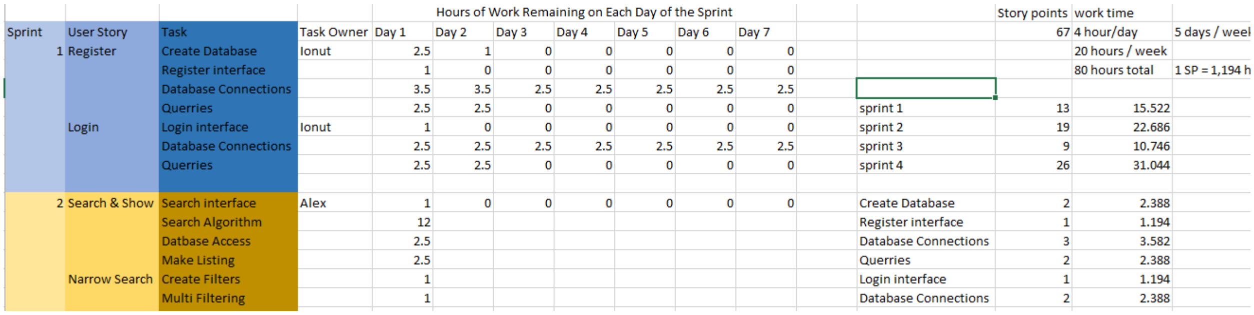
ID	Name	As a/an	I want to…	so that…	Priority	Story points	Status
1	Register	User	Be able to create an account	I can login in	1	8	done
2	Login	User	Be able to login	I can search for people	1	5	done
3	Update profile	User	Be able to update my profile picture, information and passions	people can find me	7		
4	Delete account	User	Be able to delete the account	I can remove myself from searches	7		
5	Search and show	User	Be able to search the search for other people with certain passions	I can send a Connection request	2	15	
6	Narrow search	User	Be able to narrow a search by criteria	I can find certain people(closer, far away..)	2	3	
7	Check details	User	Be able to select a person from result	I can see the details about it	2	3	
8	Send connection request	User	Be able to send a connection request	I can talk with that person	3	2	
9	Accept & Matching	User	Be able to accept connection requests	I can add a person from connection list	3	6	
10	Remove/decline request	User	Be able to remove/decline connection request	I can remove/decline a person from connection list	3	1	
11	Report user	User	Be able to report another user	I can remove spammer and annoying people	9		
12	Send message	User	Be able to message one or more people	I can interact with them	4	10	
13	Answer message	User	Be able to answer to messages	We can have a dialog	4	4	
14	Create group	User	Be able to create a group	I can message multiple people at once	6		
15	Delete group	User	Be able to delete a group	I can remove a conversation from list	6		
16	Add person to group	User	Be able to add a person to group	they can see the entire conversation	6		
17	Remove person from group	User	Be able to remove a person from a group	they don't see the conversation anymore	6		
18	Search person	User	Be able to search for a person in chat/group chat	I can private message it	8		
19	Book assistance	User	Book an advisor	I can book an advisor and ask for help	5	12	
20	Block users from app	Administrator	Be able to block users from accessing the service (ban ip and account)	I can remove fraud	10		
21	Warn users from app	Administrator	Be able to warn users if they make mistakes	I can maintain a balanced community			
22	Add premium service	Administrator	Be able to add a premium service	I can get more income	11		
23	Remove premium service	Administrator	be able to remove a premium service	I can protect the users from buying a non-existing service	11		
24	Premium service price	Administrator	Be able to change the price of the premium service	I can create discount	11		
25	Make users administrators	Administrator	Be able to allow another user to become an administrator	I can hire people that can use the app.	12		
26	Remove users rights	Administrator	Be able to drank an administrator to become a user	I can fire people	12		

Acceptance Criteria:



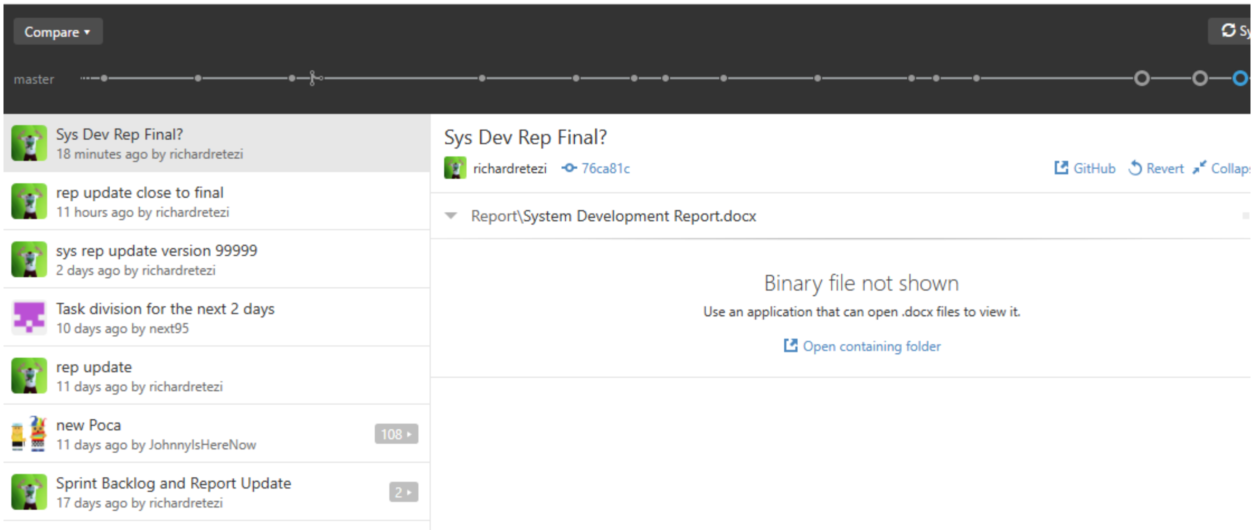
ID	How to	Acceptance criteria
1	The user will see a register button on the screen. After he presses the button he will get to a new screen where he can input his information (name, username, password, address, passios, phone no., e-mail). After he presses register he will have an account.	information is the right format, username is not taken, e-mail not used
2	The user will see two boxes where he can input his username/email and password. After he presses the login button the user is logged in into the app.	username/email exists, password is correct for that username
3	While he is logged in the user can press the My Profile button which will lead to a page with that user's information. He can edit every cell (except username) and he can press the save button to save the changes	inormation is the right format
	On the bottom of the profile page is the delete button. The user will be able to press the button and delete the account. As a verification	password is correct

Sprint Backlog



				Hours of Work Remaining on Each Day of the Sprint									Story points	work time	
Sprint	User Story	Task	Task Owner	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7			67	4 hour/day	5 days / weel
1	Register	Create Database	Ionut	2.5	1	0	0	0	0	0				20 hours / week	
		Register interface		1	0	0	0	0	0	0				80 hours total	1 SP = 1,194 h
		Database Connections		3.5	3.5	2.5	2.5	2.5	2.5	2.5					
		Querries		2.5	2.5	0	0	0	0	0		sprint 1	13	15.522	
	Login	Login interface	Ionut	1	0	0	0	0	0	0		sprint 2	19	22.686	
		Database Connections		2.5	2.5	2.5	2.5	2.5	2.5	2.5		sprint 3	9	10.746	
		Querries		2.5	2.5	0	0	0	0	0		sprint 4	26	31.044	
2	Search & Show	Search interface	Alex	1	0	0	0	0	0	0		Create Database	2	2.388	
		Search Algorithm		12								Register interface	1	1.194	
		Datbase Access		2.5								Database Connections	3	3.582	
		Make Listing		2.5								Querries	2	2.388	
	Narrow Search	Create Filters		1								Login interface	1	1.194	
		Multi Filtering		1								Database Connections	2	2.388	

GitHub



Sys Dev Rep Final?
18 minutes ago by richardretezi

rep update close to final
11 hours ago by richardretezi

sys rep update version 99999
2 days ago by richardretezi

Task division for the next 2 days
10 days ago by next95

rep update
11 days ago by richardretezi

new Poca
11 days ago by JohnnyIsHereNow
108 ▸

Sprint Backlog and Report Update
17 days ago by richardretezi
2 ▸

Sys Dev Rep Final?

richardretezi 76ca81c

Report\System Development Report.docx

Binary file not shown

Use an application that can open .docx files to view it.