FREEDOM BIKE PROJECT REPORT

by

SOLUTION 404

Murad Eminov -922154
Khaled Mosharraf -923524
Amir Doorandish -923724
Chandni Jariwala -924275
Chetankumar Dobariya -924294
Abdul Razak Shaik -926317

A Project Report submitted in partial fulfilment of the requirements for the course
Agile Requirements Engineering

FACHHOCHSCHULE KIEL
University of Applied Sciences

June 26, 2015

EXECUTIVE SUMMARY

Solution 404

Mission: Solution 404 team has to build a valid mobile device app for Freedom Bikes company customers as well as desktop oriented software for management of bikes/e-bikes renting and support for Freedom Bikes employees.

Business Description: Solution 404 is team consisting of members who will be acting as executives of different roles within Scrum Agile development methodology in order to achieve the mission.

Company Background: Solution 404 staff is consisting of 6 IT specialists who are familiar with different modern techniques, skills and experience required to carry out the mission.

Technologies: Web based solution is the main technology we've implemented. The customer oriented App proposed by us satisfies the customers of Freedom Bikes in high level. The software based solution designed for the employees on the other hand gives on-line based up-to-date information for the employees necessary to carry out important decision regarding the renting of their bikes/E-bikes as well control the bikes' maintenance.

Market: The both of our solutions (the App and the Software) are designed for bike renting purposes; however we assume that our solutions could as well provide renting other items (e.g. motorbikes or cars) along with bikes in the future if desired.

Competition: The competitive advantage of our solutions (the App and the Software) lies in their easy-to-operate interfaces as well as robustness and reliability of the systems.

Financial Projections: Please, refer to Appendix

TABLE OF CONTENTS

LIST OF FIGURES……..…..... viii

Project case statement……..……1

1 Introduction…….…….…….…….…… 2

2 Scrum Roles……..…….……..…….…....3

2.1 Product Owner……..……..…….……...3

2.2. Scrum Master…….……..……...3

2.3 Development Team…….…….…….……...4

3 Project steps……...…….….……..4

3.1 Interview people about using services…….……...…...……...4

3.2 Define Personas……...…….…..…….…….……..……..5

3.3 Empathy for Persona…….…….….…….……..……..5

3.4 User Stories……..…….…….....6

3.5 Product Backlog…….……..…….…….…6

3.6 Sprint Backlog…….……..……..……..…….…7

3.7 Planing Poker …..……..……...……..…….……...8

4 Background …...……..……..……...…….……9

4.1 Agile Requirements Engineering……..…….……9

4.2 SCRUM Methodology……..….…….…….……..11

4.3 SCRUM vs Waterfall……...….…….…….…….……..12

5 Scope and Purpose……...….…….…….…….……..16

5.1 Freedom Bikes…..……...….…….…….…….……..16

5.2 App and Software for management…...…….…….…….……..17

6 Methods and Instrumentation…….…...…….……..……..20

6.1 TASK1[Interview]……..…...…….……..……..20

6.2 TASK2[Personas]…….……..…...…….……..……..23

6.3 TASK3[Empathy Map]…… …….……...……..27

6.4 TASK4[User Stories]……...…… …….……...……..31

6.5 TASK5[Mobile App Prototype]...…… …….……..……..37
6.6 TASK6[Interview on App Prototype v.1.0]..…….……...……..51
6.7.1 TASK7[Non-Functional prototype v.2.0]...…….……...…….52
6.7.2 Non-Functional Software Protoype……..……...…55
6.8 Product Backlog……..……..……...…71
6.9 Sprint Backlog...……..……..……...…71
6.10 Roles in Scrum..……..……..……...…73
6.10.1 Product Owner..……..…….…….….…73
6.10.2 Scrum Master.....……...……..…….……..…75
6.10.3 Scrum Team…...……...…….……..…76
6.11 Events in Scrum……..…….……..…76
6.11.1 Sprint……..……..…….……..…77
6.11.2 Sprint Backlog……..…….……..…80
6.11.3 Burn Down Chart……..……..…82
6.12 Task10[Test Cases]….……..……..…84
6.13 Task11[Planing Poker]….……..……..…94
7 Our Estimations…….…….94
8 Appendix……..……..…95,96
References

LIST OF FIGURES

Fig1.Scrum Roles……. …….…….3
Fig2.Scrum Framework...……. …….…….4
Fig3.Product Backlog……... …….…….7
Fig4.Planning Poker Cards……... …….…….8
Fig5.Agile Development Overview……..…….……..…….10
Fig6.Google Trends Agile methods chart.…….……..…….13
Fig7.Agile vs Waterfall……...…….…… ……..…….14
Fig8.Mobile device usage trends……... ……..…….18
Fig9.Interview……... ……..…….20
Fig10.Persona:Amanda……. ……..…….24
Fig11.Persona:Tina……..……. ……..…….25
Fig12.Persona:Morgan…….. ……..…….25
Fig13.Persona:Tina……..……. ……..…….26
Fig14.Empathy maps obtained as result of interviews……. ……..…….30
Fig15.User story definition template…… ……..…….31
Fig16.User stories mapping……..…… ……..…….36
Fig17.Team size vs Success…….……..…… ……..39
Fig18.Non-Functional Prototype v.1.0…….……..…… ……..41
Fig19.Non-Functional Prototype v.1.0.Autheticated Access.……...…… ……..42
Fig20.Non-Functional Prototype v.1.0.Autheticated Access.……...…… ……..43
Fig21.Non-Functional Prototype v.1.0.Map view and renting ……..……..44
Fig22.Non-Functional Prototype v.1.0.One time registration……...45
Fig23.Non-Functional Prototype v.1.0.One time registration……...46
Fig24.Non-Functional Prototype v.1.0.Offline operation…….47
Fig25.Non-Functional Prototype v.1.0.Detailed view….……48

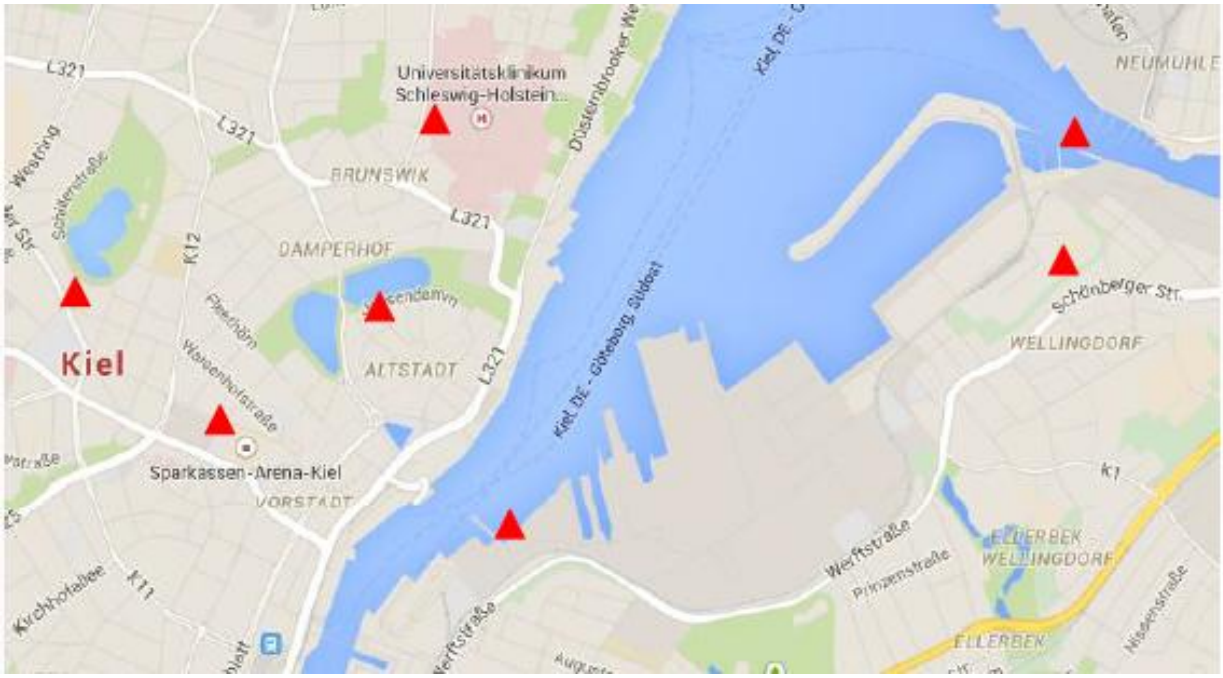
Fig26.Non-Functional Prototype v.1.0.Detailed view.Online payment……49
Fig27.Non-Functional Prototype v.1.0.Settings……..……50
Fig28.Non-Functional Prototype v.2.0……52
Fig29.Non-Functional Prototype v.2.0.Bike locator……53
Fig30.Home page of software prototype……...55
Fig31.Login windows for user……..56
Fig32.Homepage……...57
Fig33.Notification windows……..58
Fig34.Window of type of bikes with their rent charge...……..59
Fig35.Location window of software……...……..60
Fig36.Pricing window of all bikes……...……...……..61
Fig37.Customer window 1……...……...……..62
Fig38.Customer window 2……...……...……..63
Fig39.Customer chat window 3……...……...……..64
Fig40.Maintenance window...……...…...……...……..65
Fig41.Track bike window 1…..……...…...……...……..66
Fig42.Track window 2……..……...…...……...……..67
Fig43.Racing and mountain bike rent price...……...……...68
Fig44.E-bike rent price……...……..69
Fig45.Kids bike rent price……...……...……..70
Fig46.Accessories renting price.……...……..71
Fig47.Waterfall model……...……..72
Fig48.Product Backlog……75
Fig49.Scrum meeting...……77
Fig50.First release Backlog..……78
Fig51.Second release Backlog.……79
Fig52.Third release Backlog.……...……80
Fig53.First Sprint Backlog.……...……...……81
Fig54.Second Sprint Backlog.……...……...……81

Fig55.Third Sprint Backlog.…….……...…….82
Fig56.Burn Down Chart Sprint 1…….……...…….…….83
Fig57.Burn Down Chart Sprint 2…….……...…….….83
Fig58.Burn Down Chart Sprint 3…….……...…….….84
Fig59.Planning Poker Estimation…….……...…….….95

Project Case Statement:

The Company “Freedom Bikes” would like to start their business in Kiel soon. They are specialized in lending city bikes and city E-bikes - some of their bicycle stations are highlighted on the map. An important goal for the company is to design an App for smart phones allowing customers to rent a bike or an E-bike.

At the same time, the company has to design a software which gives them the possibility to manage their bikes, their bicycle stations and their customers.



1.Introduction

Scrum is a methodology for software engineering which using to prevent in software development process, and maximize abilities of software project team.
Scrum is a framework for software development process. It uses iterative model and agile methodology. Generally, scrum used in producing software.
Scrum was invented in 1986 and used in production of many software and was successful, this report will explain and introduce one case study and steps of scrum which are used for solving the problem of case the study.

Also, this report separated 2 parts:

1- Definition and overview of tasks
2- Results of each task

According to case study this project includes 11 tasks which have written below:

1. Interview people about using (E-) city bikes including the App for managing everything round about using the (E-) city bike (rent, unlock, lock,etc.).
2. Define personas based upon your interviews.
3. Develop an empathy map for personas
4. Define the user stories who you think are essential to build the software upon.
5. Create a non-functional prototype based on your user stories. Please use story boards in order to visualize business processes.
6. Interview people about your prototype.
7. If necessary: Change some of your user stories.
8. Prioritize these user stories and build the product backlog.
9. Detail respectively break down the user stories having the highest priority in order to build the (first) sprint backlog.
10. for each selected user story: Formulate the test cases for an acceptance test.

11. Apply the method “planning poker” in order to perform a relative estimation for the most important user stories.

2.Scrum Roles

In scrum, team has different members by different roles. These roles cover:[1] [Figure 1.]

1- Product owner

2- Scrum Master

3- Development team

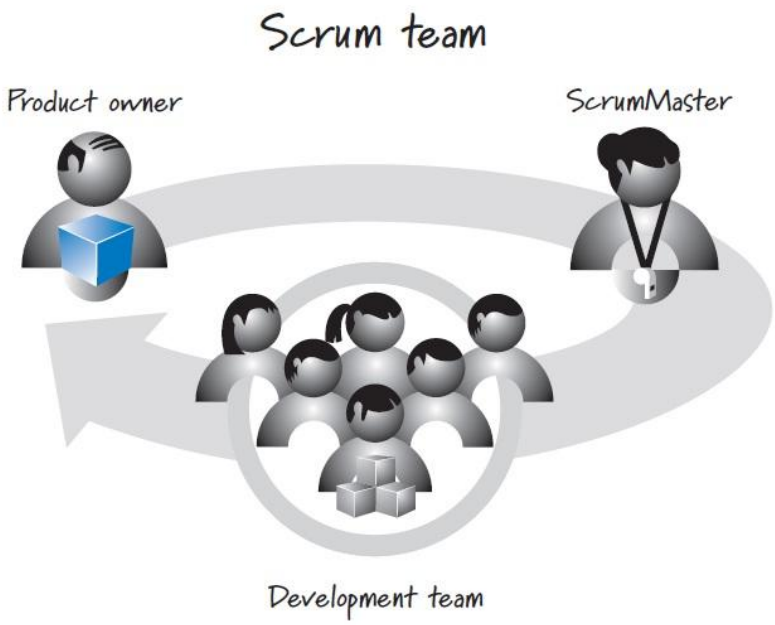


Figure 1. Scrum roles

2.1.Product owner: Product owner by exclaiming exacts requirements to team, corporate with scrum master and development team. Product owner should always be available and answer to team’s questions.

2.2.Scrum master: scrum master tasks are navigation of team members, because nobody lose framework boundary. Scrum master is not manager, he is just a leader to solve team’s problem.

2.3.Development team: team members work in scrum framework and produce project owner requirements. The number of team members depends to project are 6 till 9 person.

It's time to review scrum steps in this project.[Figure 2]

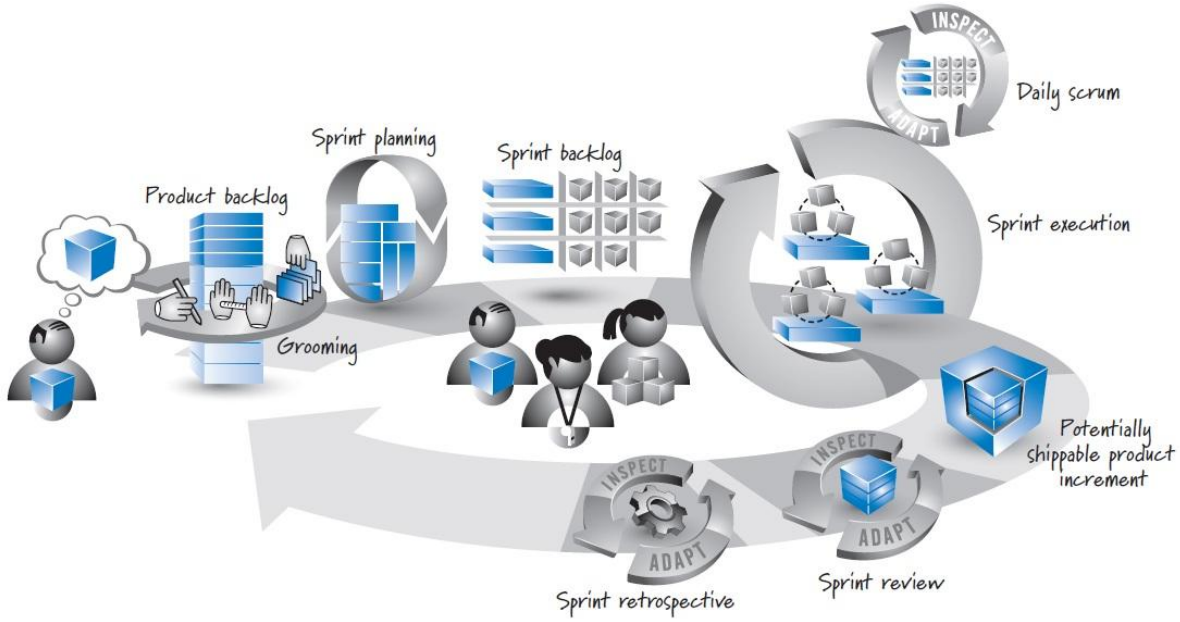


Figure 2. Scrum framework

3.Project Steps

3.1.Interview people about using services:

First step of agile process is "Interview", this task helps team members to find out user's requirement and their expectation of product or services. In this task, team will make interview with wide range of people with different condition (Gender, age, job etc.). It leads to gathering huge volume of information from many people from different level and different requirements in society.

Questions for interview designed by all team members, then they can make interview and gathering information for next steps. These information will analysis in next steps. For example, in this part, we just write two questions of interview and other questions and answers and their analysis and statistic will review in part two of report.

1- According to number of bikes limitation, do you agree with time limitation for using bikes?

2- Which method do you prefer to pay rent of bike? (Taking money from your bank account – charging your member card)

3.2.Define Persona:

These days, correct understanding of customer's requirements is very important for most of companies for providing product and services around the world.

Major Key of success is finding user's requirements and provide product on right time to cover their specific needs.

Traditionally, companies tried to develop their product by market segmentation and demographics. But now, customer's requirements gathering by persona instead of using demographic data.

What is persona?

Persona represents of users who have similar behavior patterns in using technology, buy decision, services priority and …. Behavior generally are similar between one types of people (education, gender, age …) in fact, creating persona could lead to find demographic data.[2]

3.3.Empathy for Persona:

WHY to use an empathy map

Good design is grounded in a deep understanding of the person for whom you are designing. Designers have many techniques for developing this sort of empathy. An Empathy Map is one tool to help you synthesize your observations and draw out unexpected insights.[3]

HOW to use an empathy map

UNPACK: Create a four quadrant layout on paper or a whiteboard. Populate the map by taking note of the following four traits of your user as you review your notes, audio, and video from your fieldwork:

SAY**:** What are some quotes and defining words your user said?

DO**:** What actions and behaviours did you notice?

THINK**:** What might your user be thinking? What does this tell you about his or her beliefs?

FEEL**:** What emotions might your subject be feeling?

3.4.User story:

User story is a short definition about functions which considered by user and product owner. User stories are basic resources for product backlog.

Best situation for creating this backlog is “just-in-time” when we discuss with product owner.

3.5.Product backlog:

Product backlog is a form or simple list which shows items which should be in product. (Priority is important). All members of development team or product owner could write their opinions in backlog. Basically, scrum process and agile software development work by group mind. [Figure 3]

But important point is that just product owner allows to prioritize these backlogs, because he just knows exactly about requirements. [4]

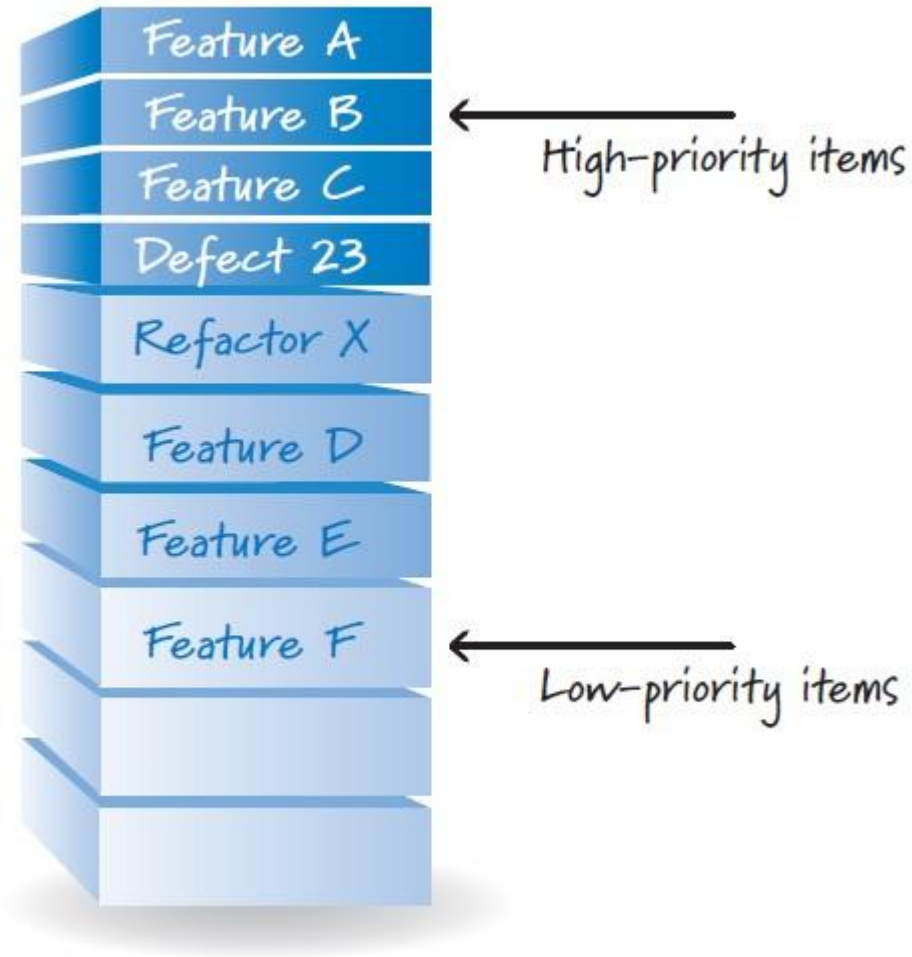


Figure 3. Product backlog

Backlog prioritizing leads to list items according to being important and waste the times.

3.6.Sprint Backlog:

Sprint is an iterative in scrum. These iterative will do 1 time in 3 weeks (2-4 weeks). At the start of sprint, some of user stories will select and will put in sprint backlog. This list includes all customer's requirements which should be implement by team in project.

Customer's requirements will save as user story (inside of product backlog or sprint backlog). The important point in user story is, it should be understandable for customer and do not in technical words.[5]

Briefly, according to speed and priority, team takes user stories from product backlog and put them in sprint backlog.

3.7.Planning poker:

The final step in our project is planning poker.

It uses for sizing each features need how many hours to implement. These features selected from product backlog. Then team members will show their estimated scale by different numbers on cards. [Figure 4]

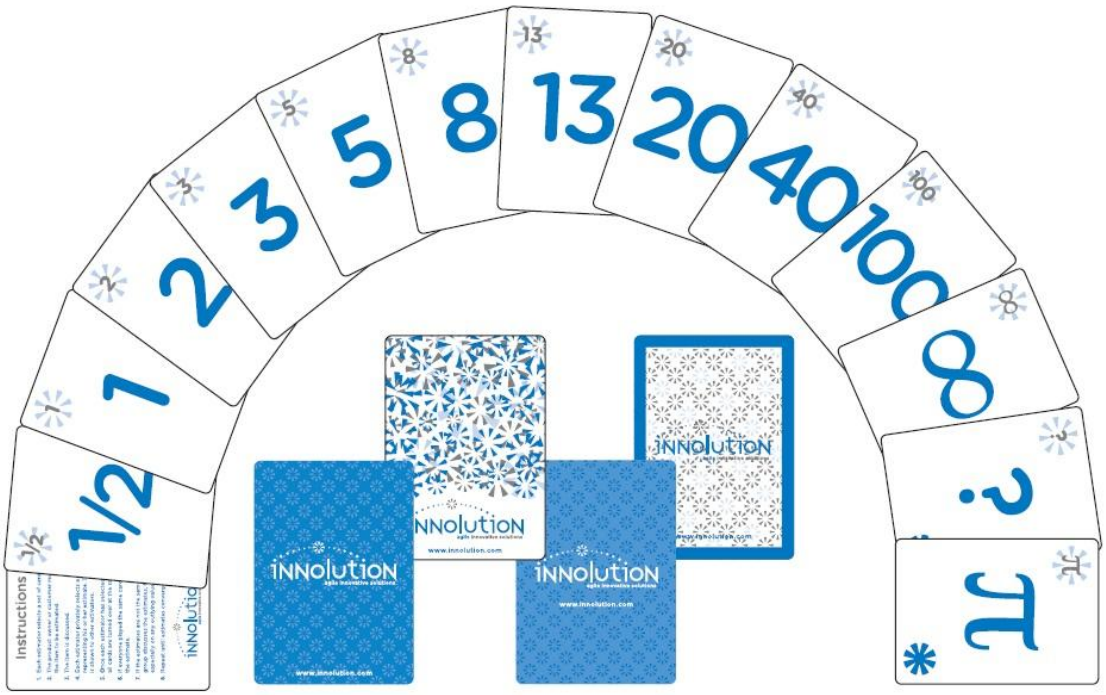


Figure 4. Planning poker cards

4.Background:

The Agile systems engineering approach leverages integrated engineering by employing the same vocabulary and artefacts in an evolving, iterative approach.Software and systems engineers work collaboratively, consider alternatives, and take actions that lead to process improvement and changes.

4.1 Agile Requirements Engineering

Agile Systems Engineering is an agile project activity that defines and develops requirements, and creates a Product Backlog for an Agile Software Development project

“Principles:

- ✓ Customer Satisfaction
- ✓ Frequent Delivery/Deployment
- ✓ Motivated Team
- ✓ Technical Excellence
- ✓ Emergent Design
- ✓ Incremental development
- ✓ Embrace Change
- ✓ Collaboration
- ✓ High Bandwidth
- ✓ Sustainable Pace
- ✓ Simplicity
- ✓ Continuous Improvement

Practices:

- ✓ Close customer collaboration
- ✓ Daily stand-up meetings
- ✓ Planning and estimating
- ✓ Frequent feedback
- ✓ Short, daily standup meetings

- ✓ Short iterations
- ✓ Prioritized requirements
- ✓ Artifact reviews
- ✓ Self-organized teams
- ✓ Retrospectives" [6]

Why?

Agile is an iterative type approach where phases (requirements/design/build/test) needed to complete a project are generally done in parallel together. There are many agile frameworks but each effectively has the same empirical structure where tasks are broken down into small tranches (planning cycles). Requirements and solutions are continually evolving and based on priority and discipline. [Figure 5.]

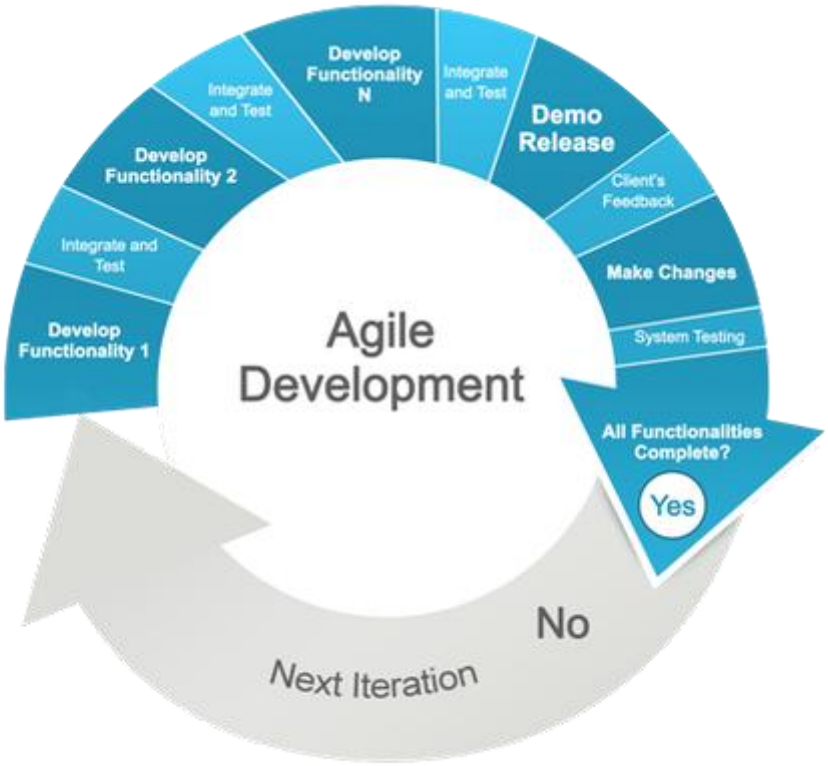


Figure 5. Agile Development overview

The advantages of Agile are:

- Change - Requirements are ever evolving so change can be catered for very easily given the shorter planning cycles.
- Encourages active involvement and interaction from key project stakeholders which allows for product build based on priority and accuracy. Continual 'show and tell' sessions allow for product agreement.
- Team interaction - Constant team involvement and structured communication channels allow for the discovery of progress, impediments and collaboration.
- Continuous improvement (i.e. lessons are learnt from and ultimately used to better the next iterations).

The disadvantages of Agile are:

- Planning (target delivery date consisting of x) can be sometimes be dubious.
- Business representation - Time and efforts will be continually required from product resources. This is crucial to cycle planning and success.
- External Stakeholder buy-in - Given its unique and alternative approach, on occasion resistance is common so training and education is needed (sometimes to a wide audience).
- Team members must be highly skilled / cross skilled in competencies as core teams are small. They must also be up-skilled on the agile framework chosen.[7]

4.2. SCRUM Methodology and its benefits

Scrum is unique because it introduced the idea of “empirical process control.” That is, Scrum uses the real-world progress of a project — not a best guess or uninformed forecast — to plan and schedule releases.

- High visibility of progress.

- Regular feedback from customer.
- Predictable rhythm.
- Measurable productivity (via burn down, velocity, etc.).
- Cross-functional, self-organizing teams.
- Inspect and adapt.
- Low bureaucratic overhead (meetings, documentation, etc.).
- Emphasis on face-to-face communication.

The model implies that you should attempt to complete a given stage before moving on to the next stage .Does not account for the fact that requirements constantly change. It also means that customers cannot use anything until the entire system is complete. The model makes no allowances for prototyping. It implies that you can get the requirements right by simply writing them down and reviewing them. The model implies that once the product is finished, everything else is maintenance.

4.3 SCRUM vs Waterfall

There are lots of ways to develop software. From the "cowboy coding" that involves no design, no documentation, and no testing to strict and rigorous processes. Almost all software development standards describe how to perform and improve specifications, designs, coding, and testing. The end goal of all software development processes are creating software that works as intended, is on time, within budget and can be maintained. But which development approach is the best. Is this context-dependent or are there any generic aspects so that an overarching software development "framework" can be labelled as "best practice".[8]

All creators and supporters of development methods, claim that their method does deliver better software, with less effort and faster than other approaches. While such an idea is nice and calming and why it is so. [Figure 7.]

The popularity (or notoriety) of Agile methods are growing and a quick Google Trends query suggests that it's not on its peak. [Figure 6.]

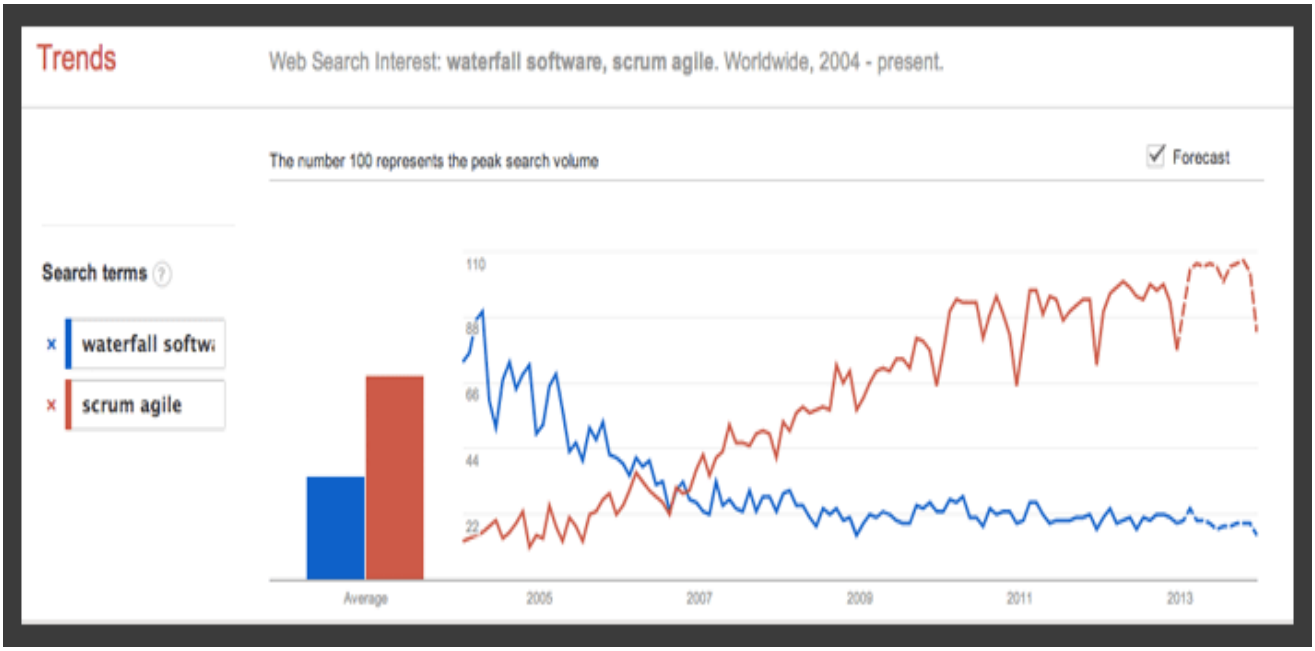


Figure 6. Google Trends Agile methods chart

However, there is one major dilemma. As much of a fan of empirical data with rigorous statistical validity, it is maybe impossible to scientifically prove that one methodology is any better or worse than any other. Software engineering doesn't have uniform scientific data to compare and back up touted methodologies such as Scrum or Agile. Projects depend on domain, knowledge and skills of the people involved, process methodology, the choices made at various phases of the project from the technology to use to the system architecture and design: It becomes very difficult to look at projects and generalize information in such a way that it is both scientifically valid and useful across the majority of projects. [9]

A sufficient definition of the development methods are needed to be able to compare and say something meaningful about them. This is completely sad when we consider the whole Waterfall method is a straw man issue and that Royce didn't described or intended to follow such a method.

Agile Versus Traditional Waterfall

Metric	Waterfall	Agile
Planning scale	Long-term	Short-term
Distance between customer and developer	Long	Short
Time between specification and implementation	Long	Short
Time to discover problems	Long	Short
Project schedule risk	High	Low
Ability to respond quickly to change	Low	High

Figure 7. Agile vs Waterfall

In general, maybe we can say that it's pretty much impossible to draw any meaningful real-world scientific conclusions because every (non-trivial) software project is unique and software development is, by its very nature, non-deterministic. But this is a fundamental and an even greater statement than we started and is crying for a foundation.

But there are definitely attempts to compare efficiency and effectiveness of software development methods. There are plenty of anecdotes that suggest that using a specific method in a specific situation has led to great success. But these kind of subjective validation is dangerous and does not in itself prove a causal link.

In the end what you measure". Agile is popular, it gives some best practices, emphasis on motivation and responsibility. But we must also be honest and acknowledge that it

is not based on real evidence that it is better and that agile is maybe a too hollow / empty concept. When there is a great product owner, then Scrum for example can be certainly faster than a waterfall approach with a bad specification document. But a bad product owner can be certainly slower than waterfall with a great specification.

In Rapid Development to consider when choosing a lifecycle methodology:

- level of understanding of the requirements
- level of understanding of the architecture
- desired reliability, risk management
- schedule constraints
- amount of process overhead
- mid-project "course corrections"
- ability to provide the customer with visibility
- ability to provide management with visibility
- sophistication of the development team and management.

Usage Purpose of different model:

Use Agile (Scrum, XP, Kanban) when:

- you want to benefit from fast feedback and burning visibility of objective data
- you don't completely understand the value and definition of what you are building
- your payoff/downside curves could vary widely
- have a team passionate about it or a coach who will help them
- have complicated project without all the experts you need or a complex project

Use Waterfall when:

- the project is simple
- the project is complicated, but you have the expertise to deliver it
- it is all you know and you have no support for change
- the upfront investment is not risky to make
- you focus your performance measures on delivery date and budget

5. Scope and Purpose

"Freedom Bikes" wants to smart phone apps where customer can make a profile and save their account information to rent the bike. All those information is confidential so it should be safe and also for the payment issue there will be some most important and confidential information of bank or credit cards. For that in this project we purposed an application for both smartphone and software where customer can rely on the Freedom bike.

5.1. Freedom Bikes

The company "Freedom Bikes" would like to start their business in Kiel soon. They specialized in lending city bikes and city E-bikes – For the customer they want to give good feature user friendly and very good service. According to the requirements of the customer they want to add map on their apps to check the bicycle stations. So in the apps or software freedom bike want to highlight there nearest stations on the map. An important goal for the company is to design an App for smartphones allowing customers to rent a bike or an E-bike. At the same time, the company has to design a software which gives them the possibility to manage their bikes, their bicycle stations and their customers.

5.2. App and Software for management

More and more of our interaction online occurs on mobile devices. Over five billion mobile subscribers – about 77% of the world's population – are online (Source: The International Telecommunication Union). This new type of simple access for consumers is catching on with a staggering amount of smartphone owners purchasing goods and services from their phone – 41% to be exact, as found in a survey

Mobile Apps are internet applications designed to run on smartphones and other mobile devices. Mobile applications help users by connecting them to Internet services more commonly accessed on desktop or notebook computers. While opportunities abound, we have identified advantages of using mobile apps for your business speed, volume of information, and advertising. We know what we want and we want it now. Most of us can't count the times we have sat waiting for our computers to take over one minute to boot up and have found our patience tested. Mobile Apps take the "wait" out of mobile. Since most, if not all, of the information is stored within the application, there's no need to wait on loading information over a slow 3G connection. Plus, what little information needs to be updated can be done in the background, allowing you to access other information while you wait on the app to load. In fact, mobile apps are becoming so functional and popular among consumers,

The biggest advantage of the native app is that it can be customized to leverage the device capabilities that enhance the user experience. The apps help you reach a wider audience by providing a multifarious presence. The mobile app helps us to connect with a huge population and results in heavy traffic. A native app of the website enables to establish a loyal and personal relationship with the end user.

Mobile application development emphasizes on developing dynamic and flexible

applications that are easy to use, providing users a better performance experience. The mobile users obtain a deep level of interactivity with the website by installing an app of the site in their iPhone/Android. By placing a brand icon of the iPhone Apps & Android Apps on the user's home screen renders a strong engagement with the user and provides faster access to data. Apps are ideal solution if one would like to provide an immersive user experience for their ecommerce store or any interactive website. Built-in native apps work on disconnected mode too. Moreover, before taking a decision between the mobile app and mobile browser, evaluate your consumer base. [Figure 8.]The way you would like to interact with your consumers is an important determining factor. With a greater proportion of population accessing the internet in mobiles, the importance of launching a mobile app for the website has become crucial. For a high quality iPhone Applications Development or an Android Application Development, choose a company that provides cost-effective and best-in-class Mobile Application Development solution.

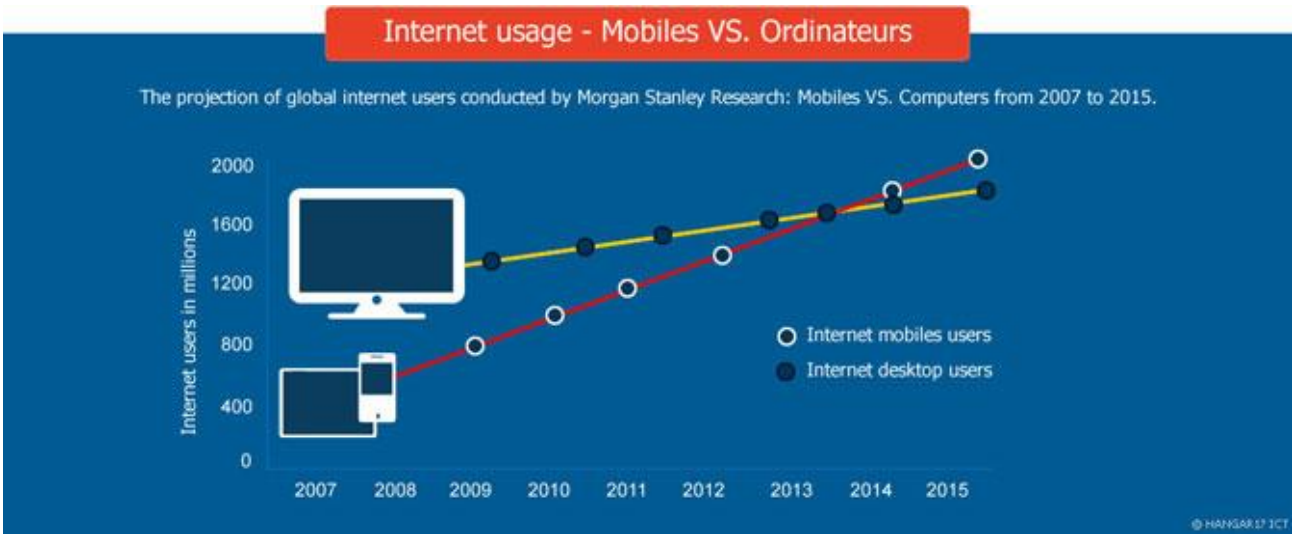


Figure 8. Mobile device usage trends

Software

Most web based applications are far more compatible across platforms than traditional installed software. Typically the minimum requirement would be a web browser of which there are many. (Internet Explorer, Firefox, Netscape to name but a few). These web browsers are available for a multitude of operating systems and so whether you use Windows, Linux or Mac OS you can still run the web application.

More manageable

Web based systems need only be installed on the server placing minimal requirements on the end user workstation. This makes maintaining and updating the system much simpler as usually it can all be done on the server. Any client updates can be deployed via the web server with relative ease.

Highly deployable

Due to the manageability and cross platform support deploying web applications to the end user is far easier. They are also ideal where bandwidth is limited and the system and data is remote to the user. At their most deployable you simply need to send the user a website address to log in to and provide them with internet access.

This has huge implications allowing you to widen access to your systems, streamline processes and improve relationships by providing more of your customers, suppliers and third parties with access to your systems.

Secure live data

Typically in larger more complex systems data is stored and moved around separate systems and data sources. In web based systems these systems and processes can often be consolidated reducing the need to move data around.

6. Methods and Instrumentation:

We as a team invested many number of hours and our hard work to complete this project successfully, which is described below in detail.

In order to design an effective application many tasks to be performed .First task to be performed is collection of interviews. Before performing this task we discussed about types and number of questions to be asked during an interview so that we can extract maximum answers form public which would be helpful for effective design of application.

6.1 TASK 1[Interviews]:

Interviews [Figure 9.] play a key role in finding the public pulse which is essential for an application to design in an effective manner.

Figure 9. Interview

Each and every one of the team performed this task of collecting interviews (5 per individual) .In total we collected 30 interviews.

The problems we faced while performing this task are:

1. Language problem where majority of people speak only German.
2. People don't have much time to spare.
3. Few are not interested in answering.
4. Few don't want their voice to be recorded.
5. It's a time consuming process as we have to move in search of public who can speak and understand English, it's one of the task that took much time, however it's the important one to develop and effective application in a smart phone.
6. Sometimes audio got recorded has lower volume which made us to make more number of interviews to gather all information again.

However we managed to get many people who are much co-operative and helped in performing this task of collecting interviews successfully.

We asked questions on basis of few concepts below:

- Availability of time.
- Location.
- Online registration.
- Personal details.
- Previous Experience.
- Suggestions.
- Ability of time.

Let's discuss few questions and answers during interviews.

Any special facility would you like to have in app for selecting desired bike?

- Few answered they need to view picture of the bike that they are going to rent.
- Few answered they need to view picture of the bike and its availability as well as rent of that.
- Few asked even about the accessories of the bike.

What about online registration?

- Some answered they are ready to online registration, and few answered its ok even to go to bike station for registration.

After booking a bike what would be your choice of delivery destination?

- Few answered it would be nice if delivery destination is nearby station, few said it's ok even if it's the main base station (like Hauptbahnhof) of bikes.

Do you need the rental service of bikes hourly or daily or weekly basis?

- Everyone gave their opinions according to their convenience like daily basis as they are tourists who need for short period like days. Few answered like hourly basis accordingly.

What type of personal information would you like to give while registration?

- Here comes different opinions like few say only name, few only name, gender and address. However they are happy to provide details and few are happy to get that registration connected with Facebook so that they need not give the information separately again, but they want the data to be secured.

How would you like to pay for your bike?

- Here also we got few answers from them like they will pay after the usage, few like half payment before and half payment after usage, few are happy to pay before usage of service.

What kind of lock would you like to prefer for your bike?

- Some answered like remote lock, smart lock, or key system.

Do you have any past experience of rental bikes? If yes then any suggestions and Opinions based on that?

- Many people don't have past experience. However few had experience in other cities.
- Had a general conversation with public and got a clear view or picture or idea how they are expecting the application to be in smart phone for their service going to launched in kiel.

6.2 TASK 2[Personas]:

- The concept of persona gives an idea to design an effective software, then it needs to be designed for a specific person.
- The aim is to bring our users to life or live environment by creating or developing personas with real names, personalities, life style, motivations and even with a photo.
- We need to develop several personas like seven to ten approximately .We develop many to make sure that we explore all the requirements of our user base.
- In general we need to follow few techniques while writing effective personas.
- Normally we don't make up personas, but instead need to discover them as a byproduct of our requirements investigation process.

- We need to know what the personas aim so that we can see what our system needs to do not to do.
- Sometimes we want to identify negative personas or people that we are not designing for.
- A first or primary persona is someone who must be satisfied but who cannot be satisfied by a user interface that is designed for another persona.
- We need a limited number of personas, our aim is to narrow down the people that we are designing for.

However there are many personas discovered by whole team, here are few personas

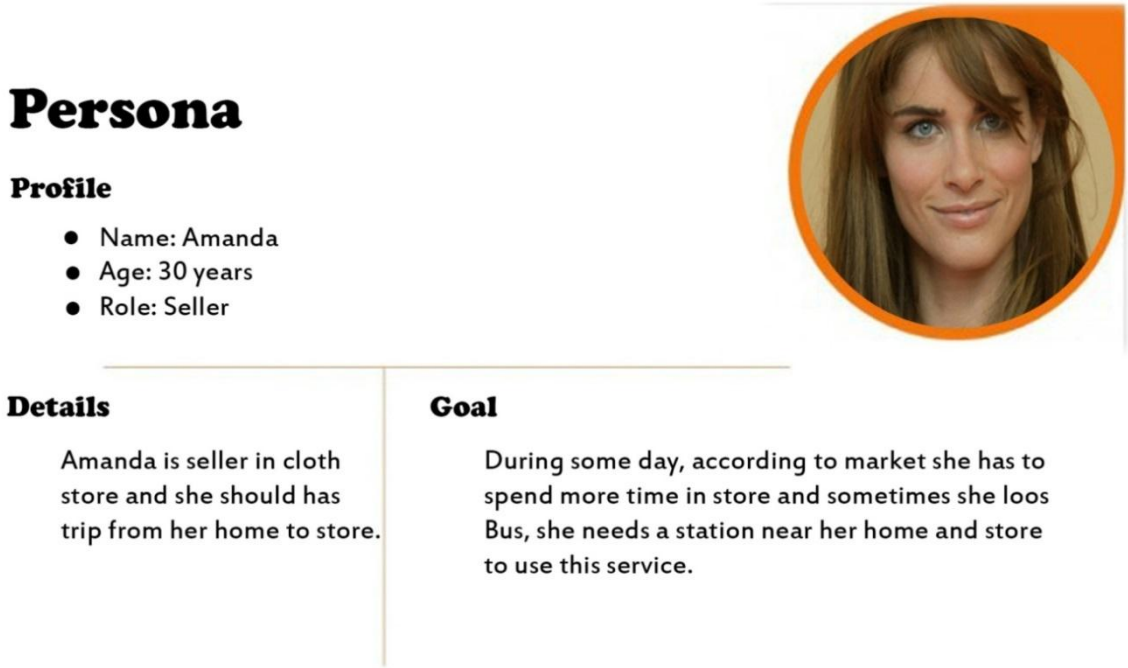


Figure 10. Persona: Amanda

Her name is Amanda. [Figure 10.] Her age is 30 who is working as a seller in cloth store, sometimes while travelling from her work place to home due to spending much time in store it becomes late and misses her bus. During this time she feels to have this service is much needed to her.

Figure 11.Persona Tina

Her name is Tina. [Figure 11.] Her age is 45 years.She is home maker or house wife.She has 3 children.She has responsibilities like buying or shopping ,going to school,sports class of her children.She prefer to use bicycle and for doing other activities during day.

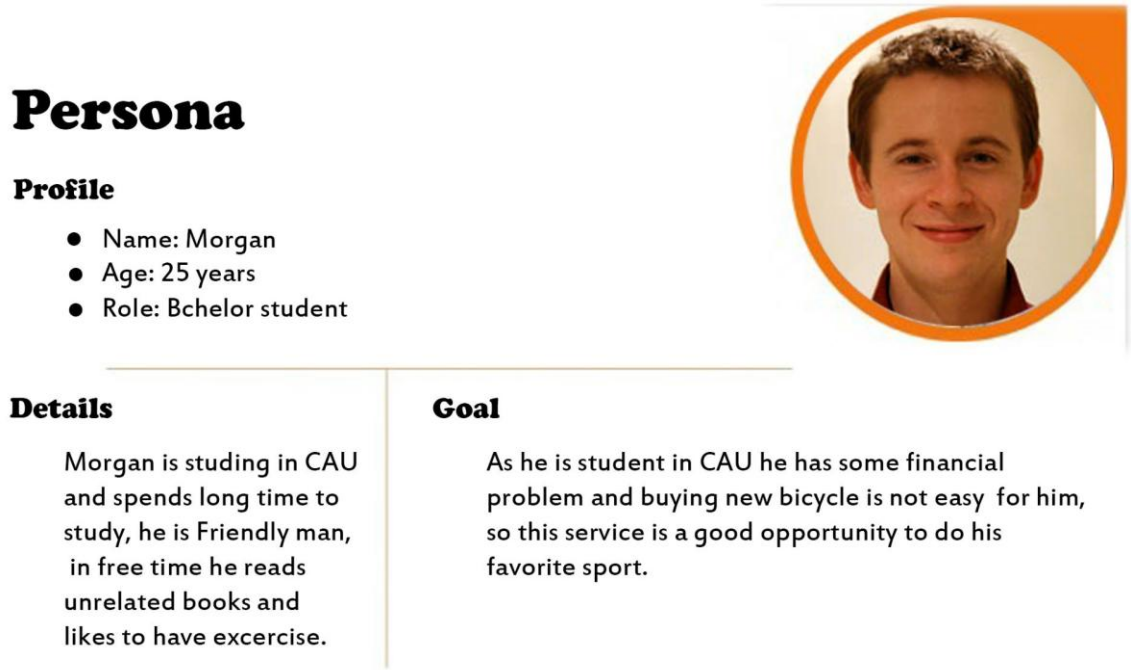


Figure 12. Persona: Morgan

His name is Morgan. [Figure 12.] His age is 25 years. He is a bachelor student pursuing his bachelors in CAU and spends long time to study. He is friendly. His hobbies are reading books, doing exercises. As he is student in CAU he has some financial problem, it's difficult to buy a bicycle, and obviously this service would benefit him for doing his favorite sport.

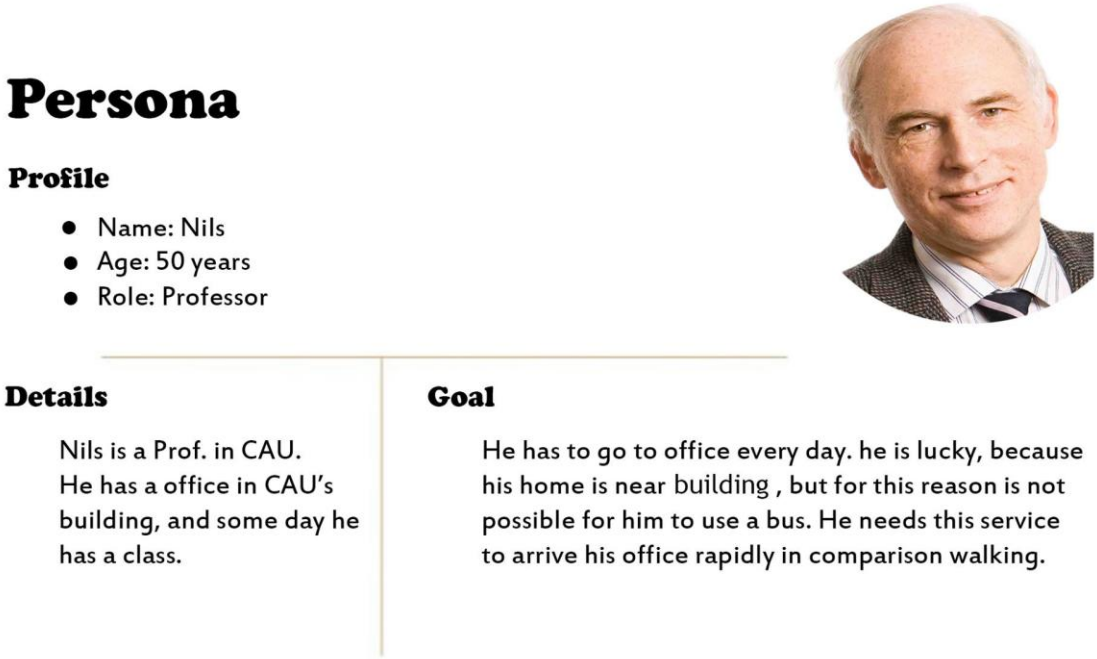


Figure 13. Persona: Nils

His name is Nils. [Figure 13.] His age is 50 years. He works as a professor in CAU. He has to go office everyday which is near to his home where this service would benefit him much better.

6.3 TASK 3[Empathy Map]:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has been gaining popularity with the agile community.

An empathy map consists of a simple face surrounded by six sections:

1. Think & Feel
2. Hear
3. See
4. Say & Do
5. Pain
6. Gain

Questions, such as what this user would be:

– Thinking & feeling about their worries or aspirations?
– Hearing while using our product, from their friends or boss?
– seeing while using our product in their environment?
– Saying & doing while using our product in public or in private?
– experiencing as a pain point or fear when using our product?
– experiencing as a positive or gain when using our product?

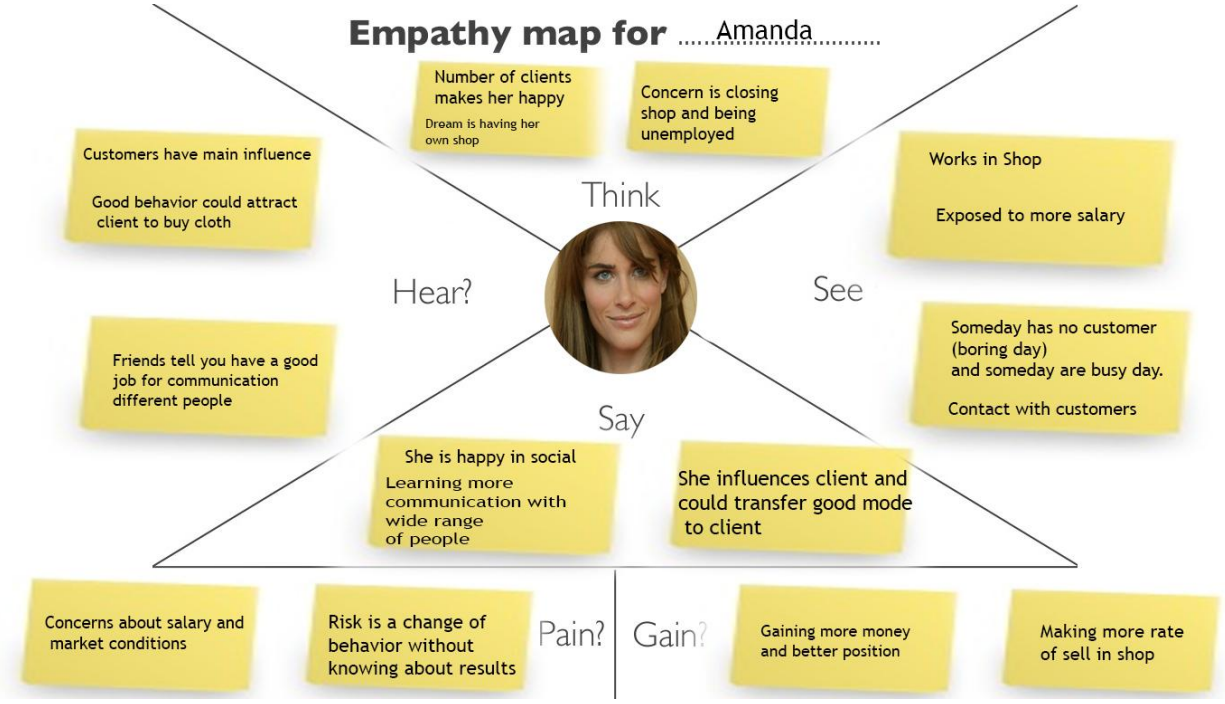
Have the team members speak about the sticky notes as they place them on the empathy map. Ask questions about deeper insights so that they can be elaborated for the rest of the team.

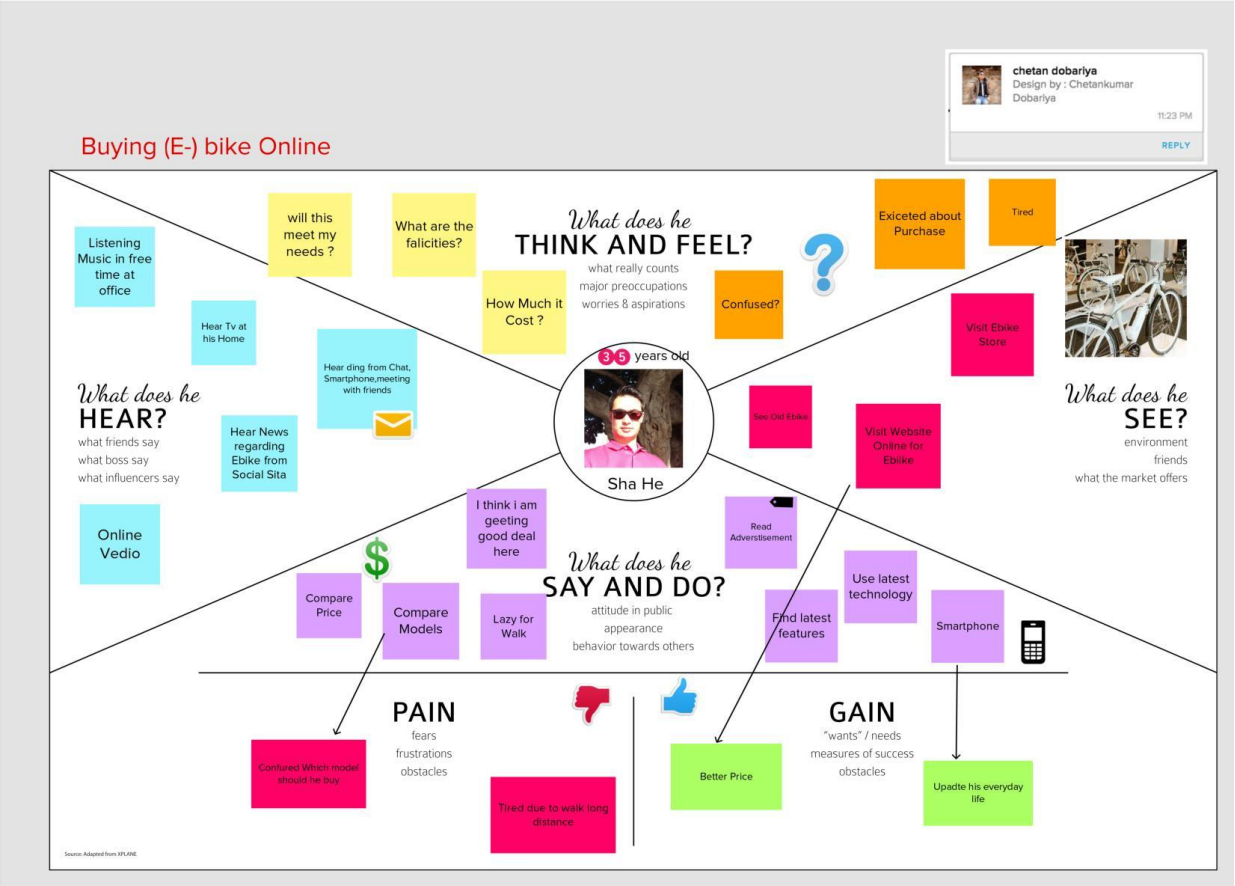
To help bring the user to life, sketch out the characteristics this person may have on the centre of the face.

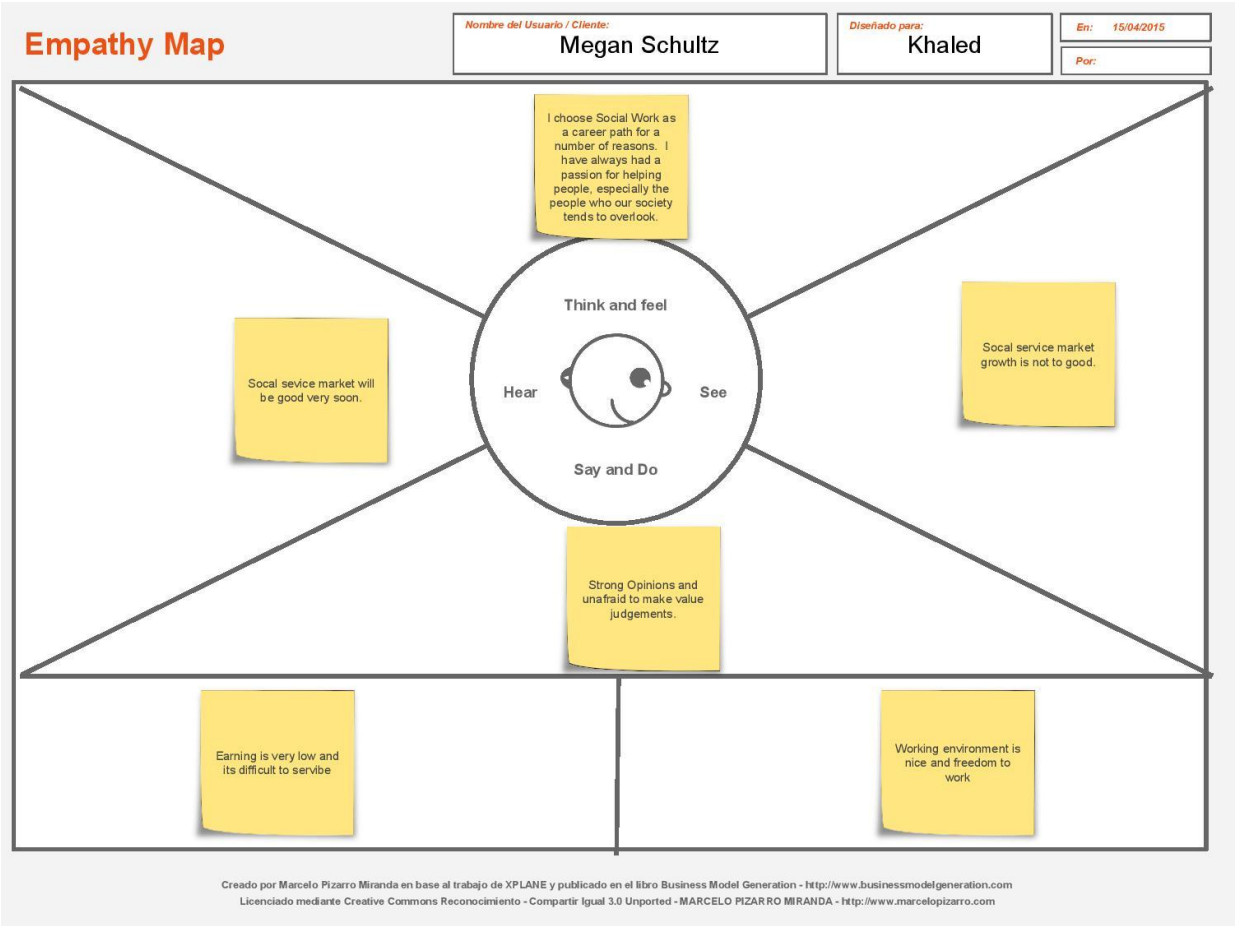
At the end of the session, ask the team members what insights they've learned. More importantly, ask them what hypotheses they now have about the users that they'd like to validate. [Figure 14.]

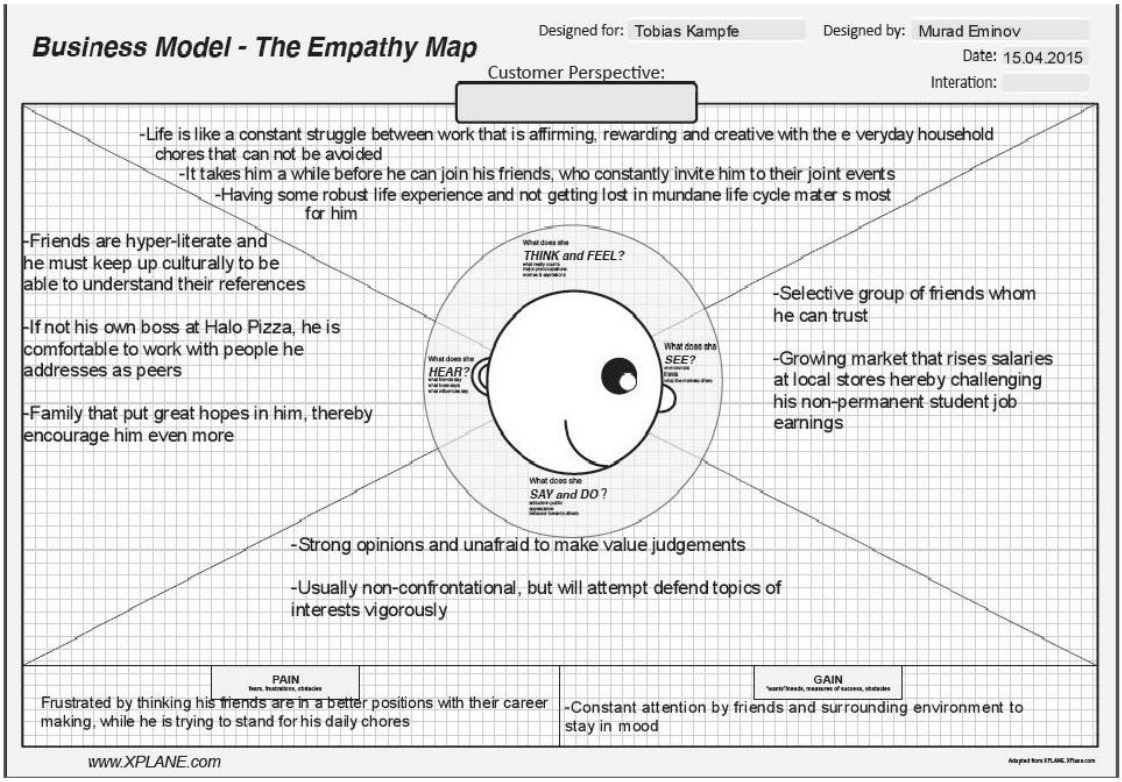
Empathy map helpful, but are not limited to:

– diving into the customer segments of a business model canvas
– elaborating on user personas
– capturing the behaviours when pair interviewing a customer
– building out the "user" in your user story.


Empathy map for .....Amanda.....
Number of clients makes her happy
Dream is having her own shop
Concern is closing shop and being unemployed
Customers have main influence
Good behavior could attract client to buy cloth
Think
Works in Shop
Exposed to more salary
Hear?
See
Friends tell you have a good job for communication different people
Someday has no customer (boring day) and someday are busy day.
Contact with customers
Say
She is happy in social
Learning more communication with wide range of people
She influences client and could transfer good mode to client
Concerns about salary and market conditions
Risk is a change of behavior without knowing about results
Pain?
Gain?
Gaining more money and better position
Making more rate of sell in shop


chetan dobariya
Design by : Chetankumar Dobariya
11:23 PM
REPLY
Buying (E-) bike Online
Listening Music in free time at office
will this meet my needs ?
What are the falicities?
What does he THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
How Much it Cost ?
Confused?
Exiceted about Purchase
Tired
Visit Ebike Store
Hear Tv at his Home
Hear ding from Chat, Smartphone,meeting with friends
36 years old
Sha He
See Old Ebike
Visit Website Online for Ebike
What does he HEAR?
what friends say
what boss say
what influencers say
Hear News regarding Ebike from Social Sita
What does he SEE?
environment
friends
what the market offers
Online Vedio
I think i am geeting good deal here
Read Adverstisement
What does he SAY AND DO?
attitude in public
appearance
behavior towards others
Compare Price
Compare Models
Lazy for Walk
Find latest features
Use latest technology
Smartphone
PAIN
fears
frustrations
obstacles
GAIN
"wants" / needs
measures of success
obstacles
Confured Which model should he buy
Tired due to walk long distance
Better Price
Upadte his everyday life


Empathy Map
Nombre del Usuario / Cliente:
Megan Schultz
Diseñado para:
Khaled
En: 15/04/2015
Por:
I choose Social Work as a career path for a number of reasons. I have always had a passion for helping people, especially the people who our society tends to overlook.
Think and feel
Hear
See
Say and Do
Socal sevice market will be good very soon.
Socal service market growth is not to good.
Strong Opinions and unafraid to make value judgements.
Earning is very low and its difficult to servibe
Working environment is nice and freedom to work
Creado por Marcelo Pizarro Miranda en base al trabajo de XPLANE y publicado en el libro Business Model Generation - http://www.businessmodelgeneration.com
Licenciado mediante Creative Commons Reconocimiento - Compartir Igual 3.0 Unported - MARCELO PIZARRO MIRANDA - http://www.marcelopizarro.com


Business Model - The Empathy Map
Designed for: Tobias Kampfe
Designed by: Murad Eminov
Date: 15.04.2015
Interation:
Customer Perspective:
-Life is like a constant struggle between work that is affirming, rewarding and creative with the e veryday household chores that can not be avoided
-It takes him a while before he can join his friends, who constantly invite him to their joint events
-Having some robust life experience and not getting lost in mundane life cycle mater s most for him
-Friends are hyper-literate and he must keep up culturally to be able to understand their references
-If not his own boss at Halo Pizza, he is comfortable to work with people he addresses as peers
-Family that put great hopes in him, thereby encourage him even more
What does she THINK and FEEL?
What does she HEAR?
What does she SEE?
What does she SAY and DO?
-Selective group of friends whom he can trust
-Growing market that rises salaries at local stores hereby challenging his non-permanent student job earnings
-Strong opinions and unafraid to make value judgements
-Usually non-confrontational, but will attempt defend topics of interests vigorously
PAIN
Frustrated by thinking his friends are in a better positions with their career making, while he is trying to stand for his daily chores
GAIN
-Constant attention by friends and surrounding environment to stay in mood
www.XPLANE.com

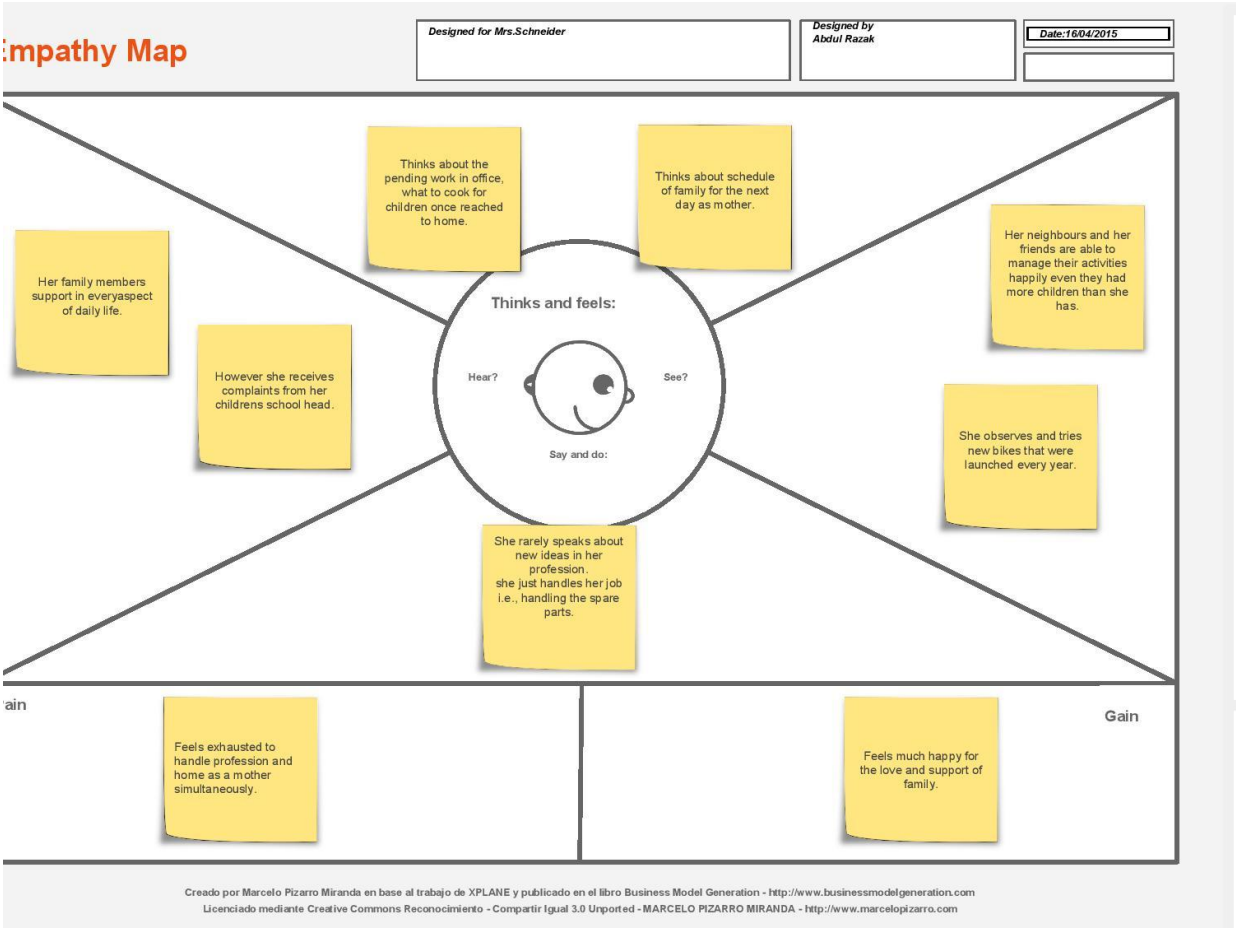


Figure14. Empathy Maps obtained as result of interviews

6.4 TASK 4[User stories]:

The heart of agile development is formed by the user stories which are short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. With an agile process you don't create all kind of documents that describe the solution from several perspectives, you just create user stories. Of course they need to be connected to the goals (as I described in an earlier post). A user story describes a change or new functionality from the perspective of a user. It will describe what the user is able to do. A user story can be initiated by an idea or a problem (bug). User stories form the connection between clients and technicians. Both of them understand functionality described in terms of what the user can do. A user story must be small enough to be realized within one sprint. This means that larger stories need to be divided into several stories, with the advantage that they remain manageable. As such, they strongly shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.

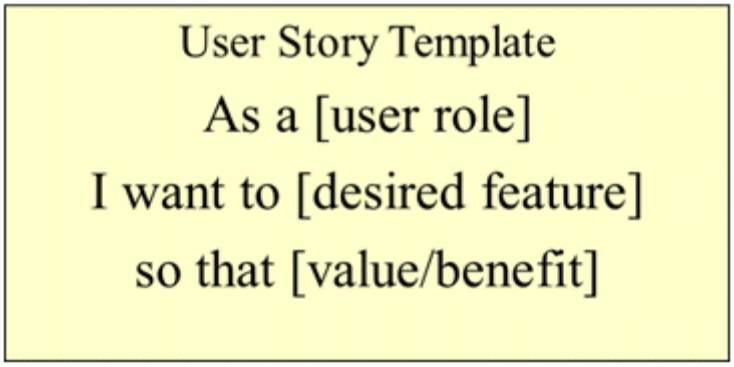


Figure 15. User Story definition Template

User stories are easy and fast to write. Clients with no software development experience at all can write them easily to communicate their goals. Not a lot of time or money is invested in writing them; therefore, if they are changed or never used, the team has not lost a lot of time in developing them.

"As a **role (w),** I want **functionality (x)** to achieve **business objective (y)**". [Figure 15.] This approach to writing user stories is regarded as the voice-of-customer approach. An example is illustrated with the images.

Characteristics of User Stories

- Stories are normally written on note cards, which contain estimates, notes or specifications on how the software should behave.
- It captures the 'who', 'what' and 'why' of a requirement in a simple, concise way, often limited in detail.

- A user story is a description consisting of one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function
- Details of the story typically come from discussions with product owners. Story writing requires teamwork. The creativity of the entire team should be leveraged in their development.
- Stories generate important discussions that reveal requirements. Stick user story posters on the wall to keep them visible and trigger discussion.
- Acceptance criteria are a way of verifying the story - they are users' conditions of satisfaction added to the user story and later used for testing the software.
- Story-writing workshops are usually organized to include developers, users, customers, etc. Participants brainstorm to generate story ideas.
- User stories are always written from the user's perspective.
- User stories are usually prioritized for each iteration. The prioritized stories are kept in what is known as a product backlog. Priorities can be changed as required.
- Stories can be decomposed starting from big/goal-oriented stories (epics) and decomposed into smaller or shorter stories. They may also be organized using themes (grouping related stories together)

Mistakes to Avoid in Writing User Stories

- Don't assume there's only one user. Users may vary and may not have the same goals in using a system. When roles are used, users become more tangible and it's easier to distinguish between the different user categories.
- Instead of asking users what their stories are, involve them to create a participatory design environment; this has significant benefits.
- Not everything to do with the development of the system should be captured in the form of a user story. For example, defects and design ideas don't need to be expressed as user stories.
- User stories should not be vague, informal and incomplete.
- The user stories should not solely rely on user stories

INVEST in Good Stories

Bill Wake introduced an acronym known an **INVEST** for writing user stories. Stories should have the certain key characteristics. They should be:

Independent - Good stories should be independent; this facilitates effort estimation and prioritization. The developer should be able to select a story without pulling in other stories. With this quality, features of a story can be built and demonstrated on their own, without depending on other stories.

Negotiable - Stories are not binding. They are not contracts and should be approached with a reasonable degree of flexibility. Stories may thus be split into several mini-stories and the evaluation criteria should also be relaxed where necessary.

Valuable – Stories should be valuable to customers or the real users of the system; they should deliver demonstrable business value as quickly as possible.

Estimable - Plans for the project are usually based on user stories; stories should contain information that helps the developer arrive at a fair estimation of the duration of time each will take for completion.

Sized Appropriately - Complex stories imply that users will have to wait a considerably long time before value can be delivered. One thing to note is that if the stories do not fit on a card, the scope may be too large. As common practice, user acceptance criteria are written on the back of the card.

Testable - Stories should be testable. Acceptance criteria are used to verify the functionality of the software.

Productive Collaboration - User stories help keep the client and development team on the same page for the requirements of the project. This leads to more productive collaboration between the parties to make the delivery process more efficient.

Most user stories will not have all these qualities from the beginning and will need to be constantly revised until they do.

According to our case User story.

Stake holder name: - user stories:

Customer:

- As a consumer, I would like that my bike integrated with smart phone, so that, I can lock and unlock bike easily.
- As a consumer, I would like that my bike has smart phone app integration, so that, I can see my daily usage of energy assumption.
- As a consumer, I would like to get price range and features to get bike. So that, I would choose bike accordingly.
- As a consumer, I would like to get bike for daily and hourly bases, so I can manage rent of bike according to my usage.
- As a consumer, I want that software should keep my personal data confidential and secure.
- As a consumer, I like that software provide online transaction facility to rent e-bike, so that I can easily make transaction to rent e-bike.

- As a Consumer, I need online chat system in software, so I can communicate directly and immediately and fulfil my requirement.
- As a consumer, I expect an application in my smart phone which would save my time.
- As a consumer, I like to have the complete overview of the product so that I can select them accordingly
- As a consumer, I like to have facilities like GPS in product.
- As a consumer, I like to ensure safety of the product by using smart system for locking
- As a customer I want a multi-language supported app for the foreigner so they can operate in their own terminology.
- As a customer I want Membership and Discount offer options in my smart phone app so I can save my money.
- As a website visitor, I can see all the necessary information on e-bikes so that, I can rent one
- As a consumer, I expect the services to be open throughout the whole day
- As a consumer, I can order more than one bicycle, so that I can hang out with friends
- As a consumer, we can use GPS system, so that we know our location

As a consumer, I like to ensure safety of the product by using smart system for locking

- As a consumer, I can use any payment means I have, so that I can pay for renting

Supplier:

- As a Supplier, I need the software should be flexible enough to provide user information so that I can make supply accordingly.
- As a Supplier, I would like that administrator give me base location and number of bike so that I can provide stock accordingly.
- As a supplier, I have to manage products (bikes) to all the branches accordingly.
- As a supplier, I like to take survey from the customers to understand their needs, and demands to improve the services.
- As supplier, I would like to know the number of customers, so that I can supply products accordingly

Owner:

- As an Owner, I like that software should be user friendly , so I can reach maximum number of customer
- As an Owner, I need that software should have feedback system, so I can

reached to customer requirements.

- As an owner, I like to make the services reachable easily to customers.
- As an Owner, I like to develop my business to high standards
- As an Owner, I would like to be contacted by my partners, so that I can improve my company
- As an Owner, I like to make the services reachable easily to customers

Development Team:

- As a developer, I need clear and appropriate user requirement so that I can develop software accordingly.
- As a Developer, I would like to get user requirement changes early stage of s/w development so that I can manage my time schedule for development accordingly.
- As a developer, I need to know whatever the features customers are expecting before I start the work of developing a software.
- As a developer, I expect the whole team should be aware of the needs of customer exactly so that no mistake was done while developing a software, because while developing a software all features would be inter-related.
- As a developer, I need to develop new ideas to make the application view better, attractive and useful.
- As development team, we need to be able to contact the supplier, owner and so that we can question them

- As a developer, I need to know whatever the features customers are expecting before I start the work of developing a software.

Project Manager:

- As project manager, I would like that developer should be fast and efficient in development process, so I can meet s/w delivery date accordingly.
- As project manager, I want that developer should interact in period of time during development process, so that I can resolve serious issue in time.

All those user stories we priories user story for backlog and we mapped our user story according to the importance which is give below. [Figure 16.]

User Story Mapping:

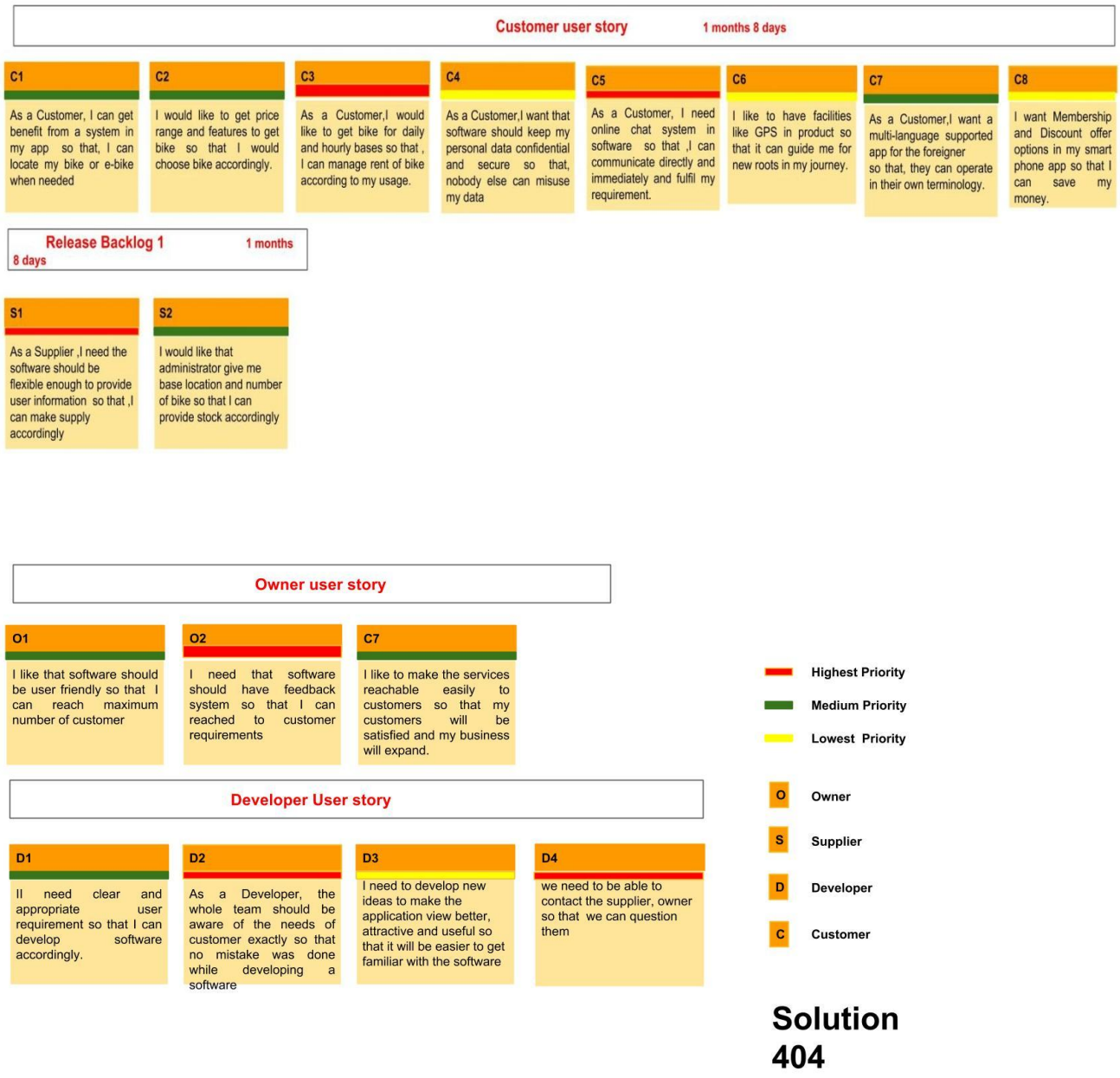


Figure 16.User Stories Mapping

User stories are not static in time but need to be expanded, negotiated and amended throughout the course of the project. The user story concatenates the viewpoint of different users and as such, may serve different purposes to each user category. It is not purely technical or purely business-oriented; this is why Roy Philips refers to it as the “boundary object”.

User stories are used with agile software development methodologies as the standard for defining the functions a business system must provide, and to facilitate requirements management.

Are user stories better than other types of requirements specification? It depends. In the end, it's the individual or team that makes a particular technique work (or fail). If a team is used to an iterative approach, use cases and user stories will be relatively easy to work with. If a team is steeped in waterfall and "big requirements up front (BRUF)," that team likely will bring that style to user stories and end up with traditional requirements that are user stories in name only. Adopting the underlying philosophy (concepts like collaboration and just-in-time requirements) is the key to moving toward more agile specification devices, such as user stories.

User stories are not a highly documented series of requirements but rather a reminder to collaborate about the topic of the user story—in other words, in agile development (good agile at least), the documentation is secondary to the collaboration. The intent of user stories is to foster collaboration; the perspective is on customer value. If your user stories look more like requirements in disguise, you need to worry less about what specification you are using and more about how to increase collaboration. If, on the other hand, you have a collaborative environment already, user stories will enhance the collaboration more than use cases and traditional requirements can, in which case user stories are a better tool for you.

6.5 TASK 5[1. Mobile App Prototype]:

In Agile Requirements Engineering, there are plenty of distinct methods implemented. Many of those requirements involve the following requirements gathering which we considered while collecting the information from the customers [10]:

Involvement of customer(s): very high importance is given to interaction with the customer. First of all, all the relevant stakeholders have to be identified. The key issue

is finding the customers for whom the final product will be made for, identifying and prioritizing their requirements is a key. Different practices help to get as much information as possible from all involved stakeholders.

Contracts for Agile Projects: Firstly, the most critical requirements are expressed by the stakeholders so that the project leaders who have high enough experience are able to figure out the initial cost for the agile project and make a guess on the cost for the later alterations.

Agile Releases: agile delivering parts of the system gives possibility to release faster the results of expectation to the customers so that a feedback could be obtained from them. However, all the requirements implement iterative and incremental fashion.

Requirements Elicitation Language: Usage of sophisticated linguistic methods in order to retrieve the information from the customer. Natural Language Processing (NLP) allows gathering the information from the customer not using formal language of specifications.

Non-Functional Requirements (NFR): In our project, we paid a high attention to defining the non-functional requirements. Once the NFRs are identified, our team began with development.

Flexibility in small agile teams: A small agile team like ours (6 people) had the comfort to easily communicate with the stakeholders (customers) in efficient way, and Control the requirements changes in agile manner. In figure 1, from [1], it is given the chart representing the dependency between the number of people in our team and the chances for success. As it can be seen from the chart, the dependency between the number of team members and the chance for the success is inverse (i.e. less the team, bigger the success). Since, our team consisted of 6 members; we estimate our project success high enough. [Figure 17.]

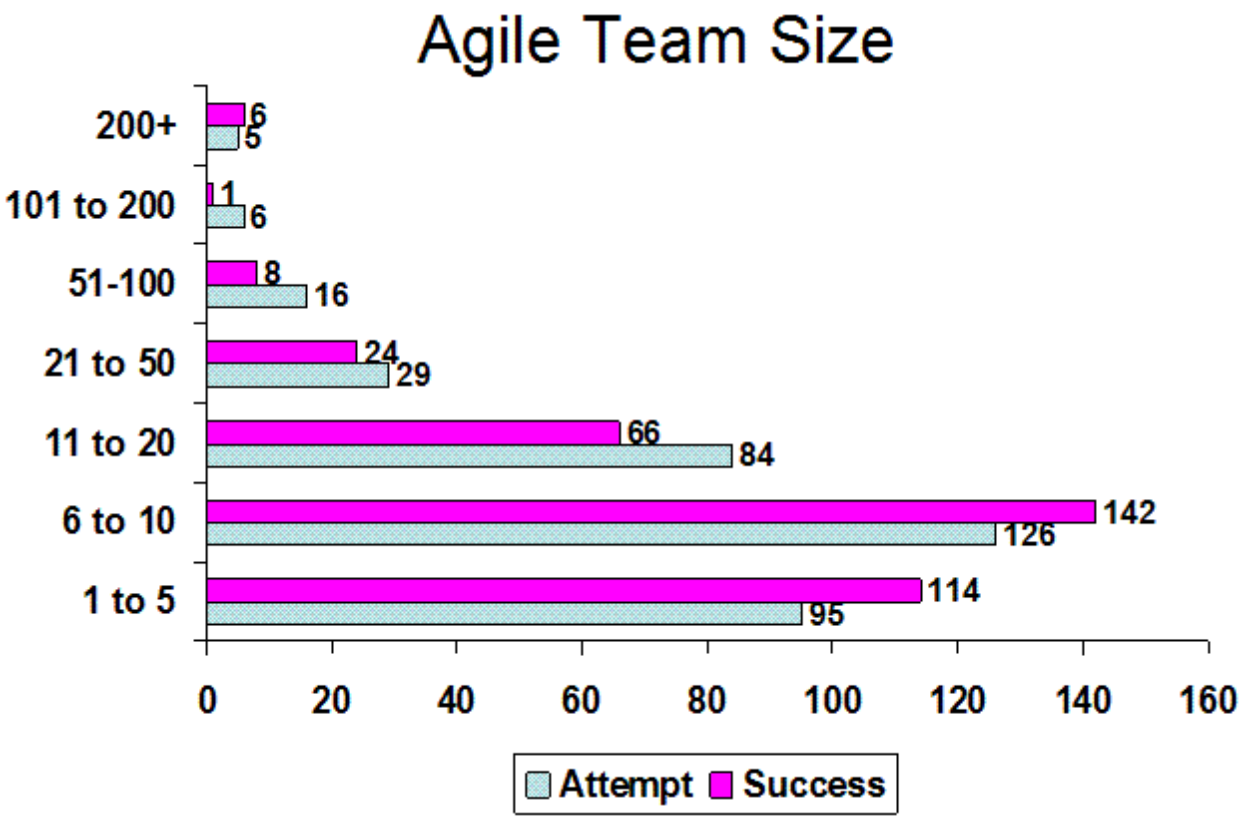


Figure 17. Team size vs. Success

Requirements splitting: We asked the customers to split the high complexity requirements into simpler part. This complexity reduction technique allowed us to better understands what functionalities are required by the final customers.

Mobile App Prototype

As all the user stories collected were collected and analyzed, and it was obtained a general insight of what product might interest the customers, a non-functional prototype was created. In general, prototyping is a process where a developer can present his/her conceptual design of final product before too much time and effort was spent to release a functional product and it appears to be not what it was initially expected in the end. Prototyping is a widely used practice, and has several benefits:

- The software designer and implementer can get valuable feedback from the users early in the project
- The client and the contractor can compare if the software made matches the software specification, according to which the software program is built.
- It also allows the software engineer some insight into the accuracy of initial project estimates and whether the deadlines and milestones proposed can be successfully met (was especially useful during Planning Poker phase)

Our non-functional prototypes were especially helpful for our customers to express their opinions on proposed solution, as well for us. We benefited from the non-functional prototypes, when our team wrote Test Cases. As all of our team members were allocated roles within our team, the prototypes simplified writing Test Cases for scenarios on perspective of diverse roles within our team. The team members could make judgments based on the proposed prototypes and considered their real-life skills, thereby the time required for completion of the tasks.

Considering the initial requirements from the user stories, the Solution 404 member **Murad Eminov** developed the app prototype below [Figure 18.]:

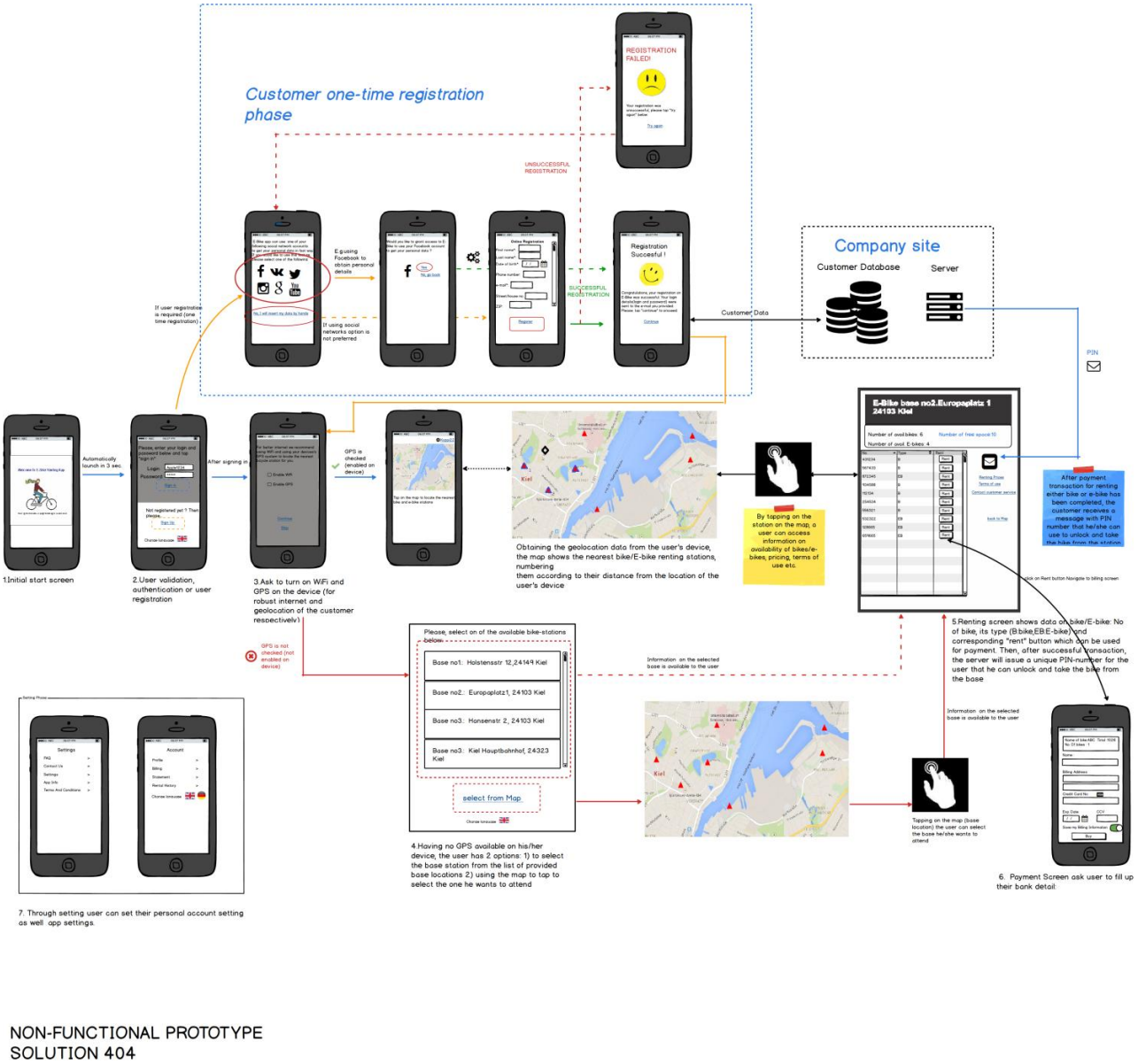


Figure 18.Non-Functional Prototype v.1.0

As it can be seen, the proposed app prototype, the mockup was made in simple and easy-to-understand manner and deprived any high-level functional or coding description that might confuse the customer.

In order to describe in more details the app prototype, we consider the fragment which illustrates the phase where the user is registered in the system and has authenticated access using two-way authentication method (login and password), [Figure 19.].below:

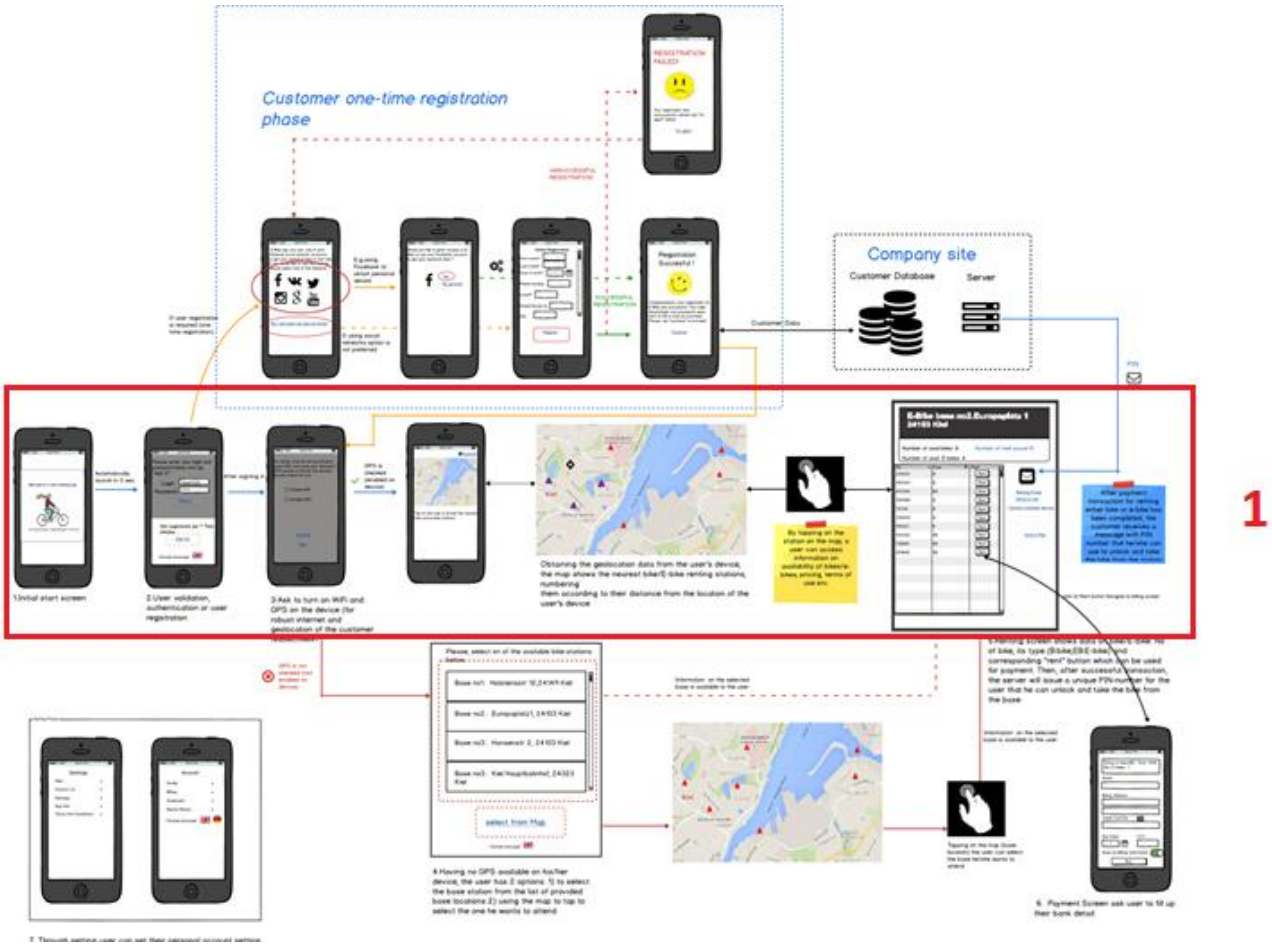


Figure 19.Non-Functional Prototype v.1.0.Authenticated access

As it can be seen from the [Figure 20.]. Below, the app launches with displaying initial screen with the logo designed for Freedom Bikes. In 3 seconds, the login page requesting the user to enter his/her credentials (login/password) appears.

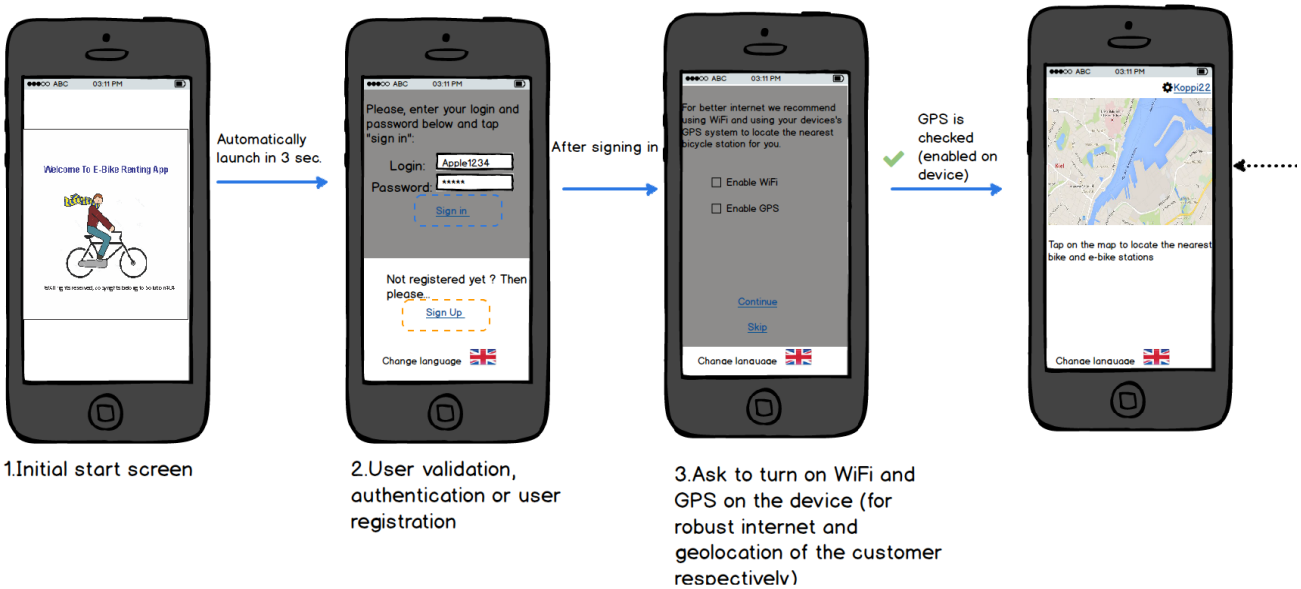


Figure 20.Non-Functional Prototype v.1.0.Authenticated access

In case of successful entering the credentials, the user taps Sign In button, where the systems verifies him. The user has option to switch among multiple languages since the beginning using the app. This is a property to allow easily adoption of the app to the global market along with the local through providing easy access to individuals with different lingual background.

In next screen, the user sees two ticks – Enable WiFi and Enable GPS. User may choose to enable the internet and GPS on his/her device or not. The app can be used in two modes: online and offline. For now, we assume that we are using the app in online mode (with internet available on the device) as well as the GPS on the device is available as well.

Thus, ticking the both options, the user sees an interactive map of Kiel city as shown below in the [Figure 21.].

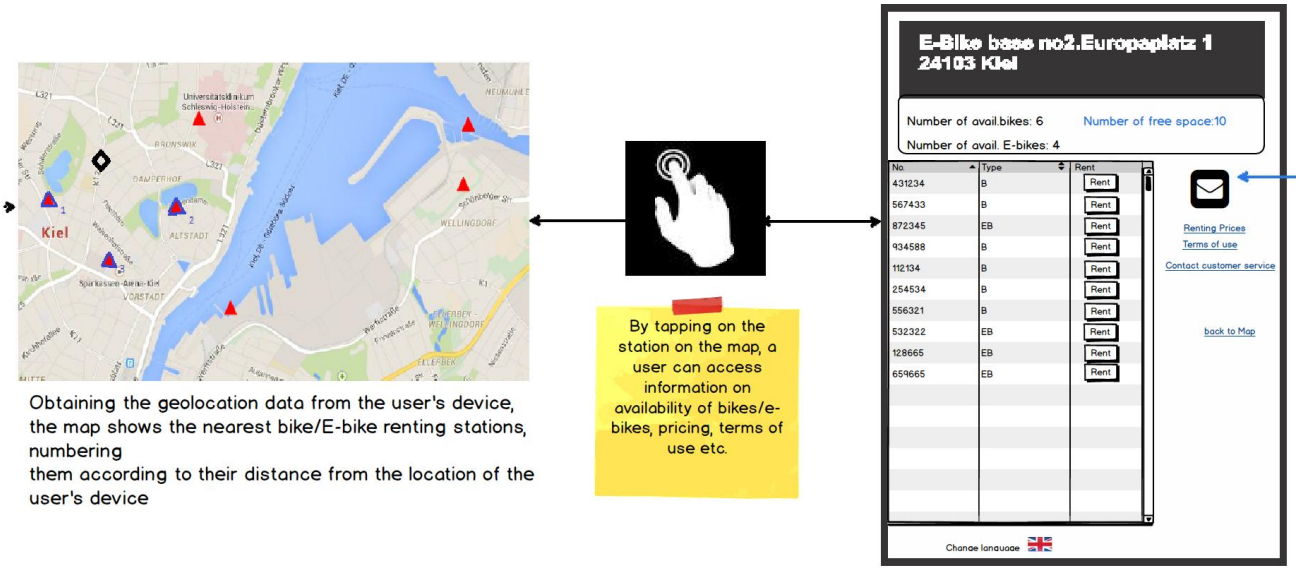


Figure 21.Non-Functional Prototype v.1.0.Interactive map view and renting

In Figure 21. It is shown how the user interacts with the map. As it can be seen, the base locations are marked as red small triangles, and spread around the Kiel city. The black marker at the north-west is the current location of the user's device. The geo-location is obtained from the user's device.

Around the marker it is possible to see triangles with blue colored sides. The three blue shaded red triangles are the base stations most close to the user's device. They have index numbers below them like 1, 2 and 3. The numbers correspond to the distance proximities from the user's device to the base station. 1 is the closest, 3 is the distant.

Considering his location and selecting the desired station from the map, the user taps on the base station. Next, a screen with base station data (such as base station name, address) appears. This screen also contains the list of bikes or e-bikes (with their IDs) that a user may consider to rent. This page also contains private message box where the user receives PIN numbers, as well as messages from the Freedom Bike. Below the message box, the Pricing Info, Terms of Use as well as Contact Customer Services elements are located.

On this page the user sees all the information regarding the number of available bikes and e-bikes he can rent. Also, he sees the info on the available empty slots for bikes

where he can return his/her bike. The user has an option to return to the Map and select another location if he/she doesn't want to use this particular bike station.

In [Figure 22.]. Below it is shown the One-Time-Registration phase, which is necessary to be held at least once by the user in order to be verified by the Freedom Bikes system. One-Time-Registration makes it possible for the Freedom Bikes Company to easily identify the user in case if any damage, loss etc. were sustained to the bike rented by this particular user (i.e. for the legal issues).

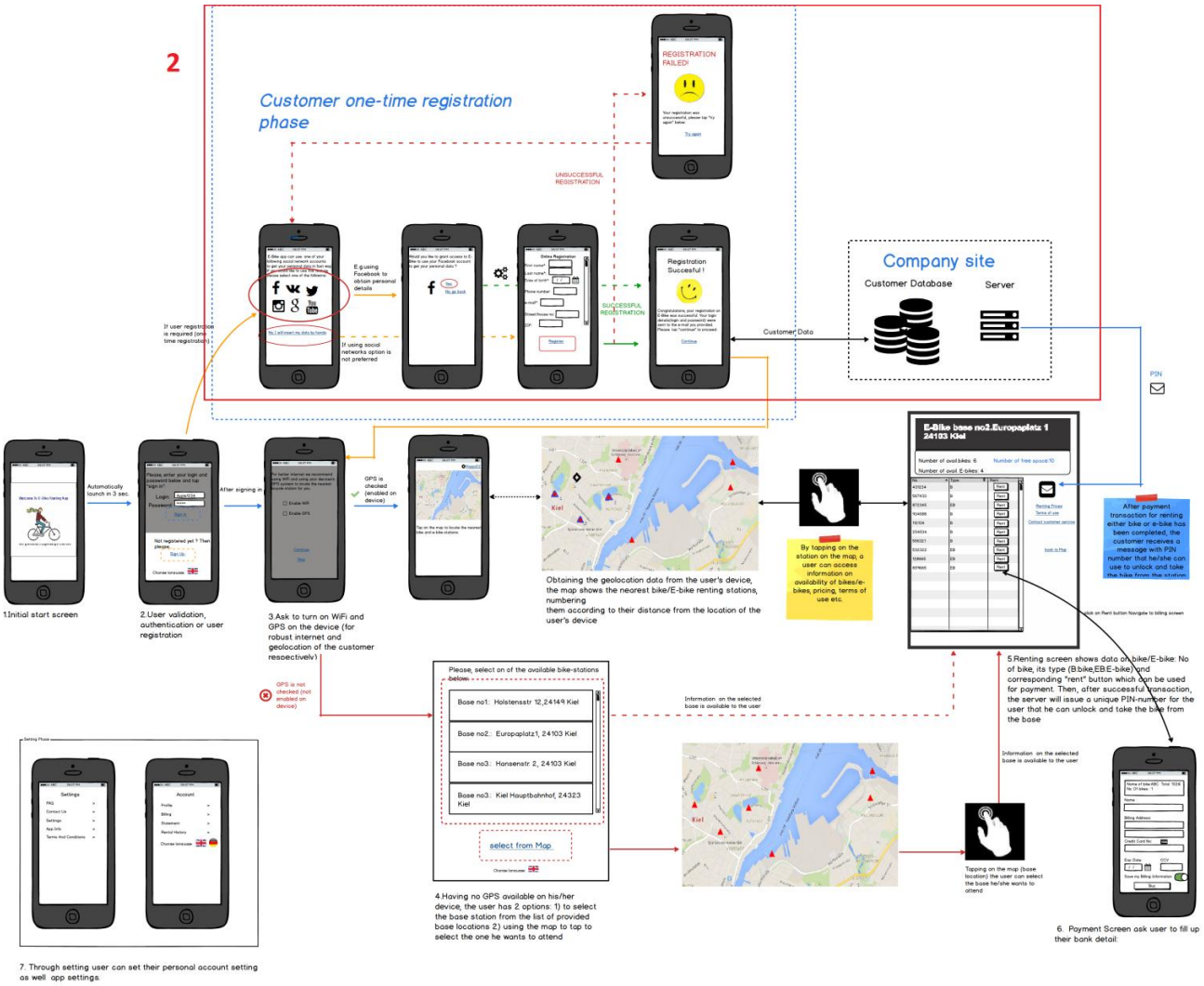


Figure 22.Non-Functional Prototype v.1.0.One-Time-Registration

The One-Time-Registration is a process where the both frontend and backend of the system are involved. The frontend is designed in a way that offer maximally comfort to the user in registration. A more detailed view of One-Time-Registration is given below in the [Figure 23.]

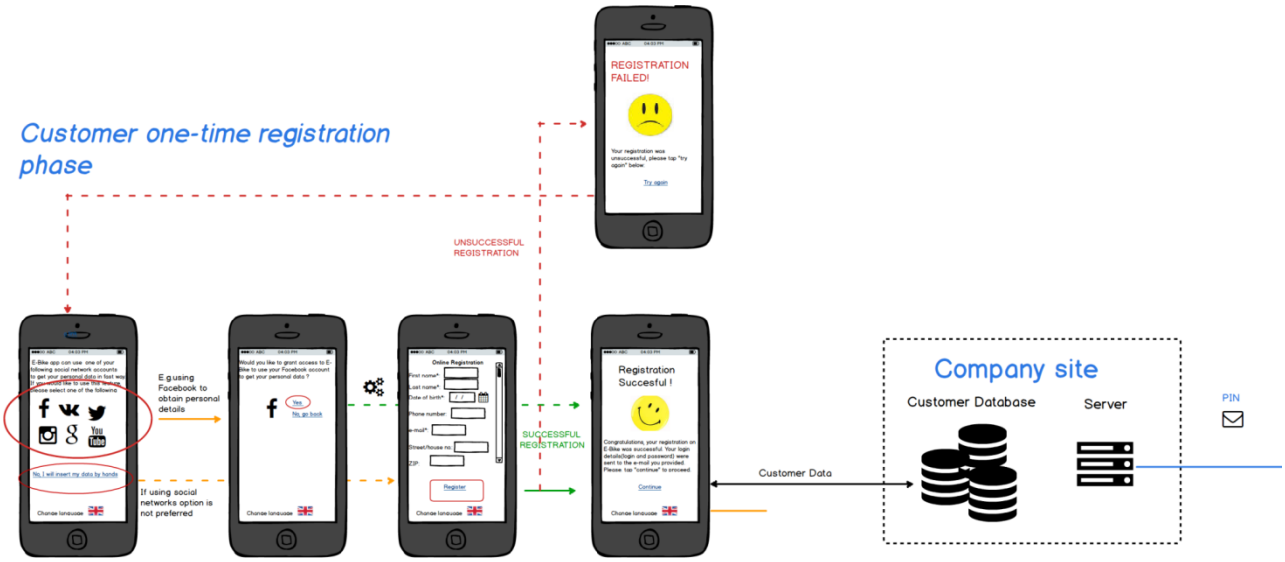


Figure 23.Non-Functional Prototype v.1.0.One-Time-Registration detailed view

As can be seen in the Figure 23., during the registration a user sees a screens where he/she is has a possibility to either use one of their social networks for fast and easy registration in backend manner, or not. If user prefers not to involve any of his/her social networks in the transaction, he/she will have to enter the user registration details using the display keyboard of the device.

Next, the user needs to tap “Register”. If the user information provided was complete and accurate, without any missing must-be data, the user sees a screen with smiling icon, indicating the successful registration. The backend customer database is now altered with a new customer record.

In case if the user entered invalid, inaccurate or incomplete user data, a sad-faced icon appears, meaning that the registration failed, and the user can try register again by tapping “Try again”.

The app can function as with device GPS on or off. If the user prefers not using his/hers device GPS (for example, for the purposes of anonymity or not willing to be disclose his location) he/she still can use the app. In Figure 24. Below it is shown the phase of offline operation of the app.

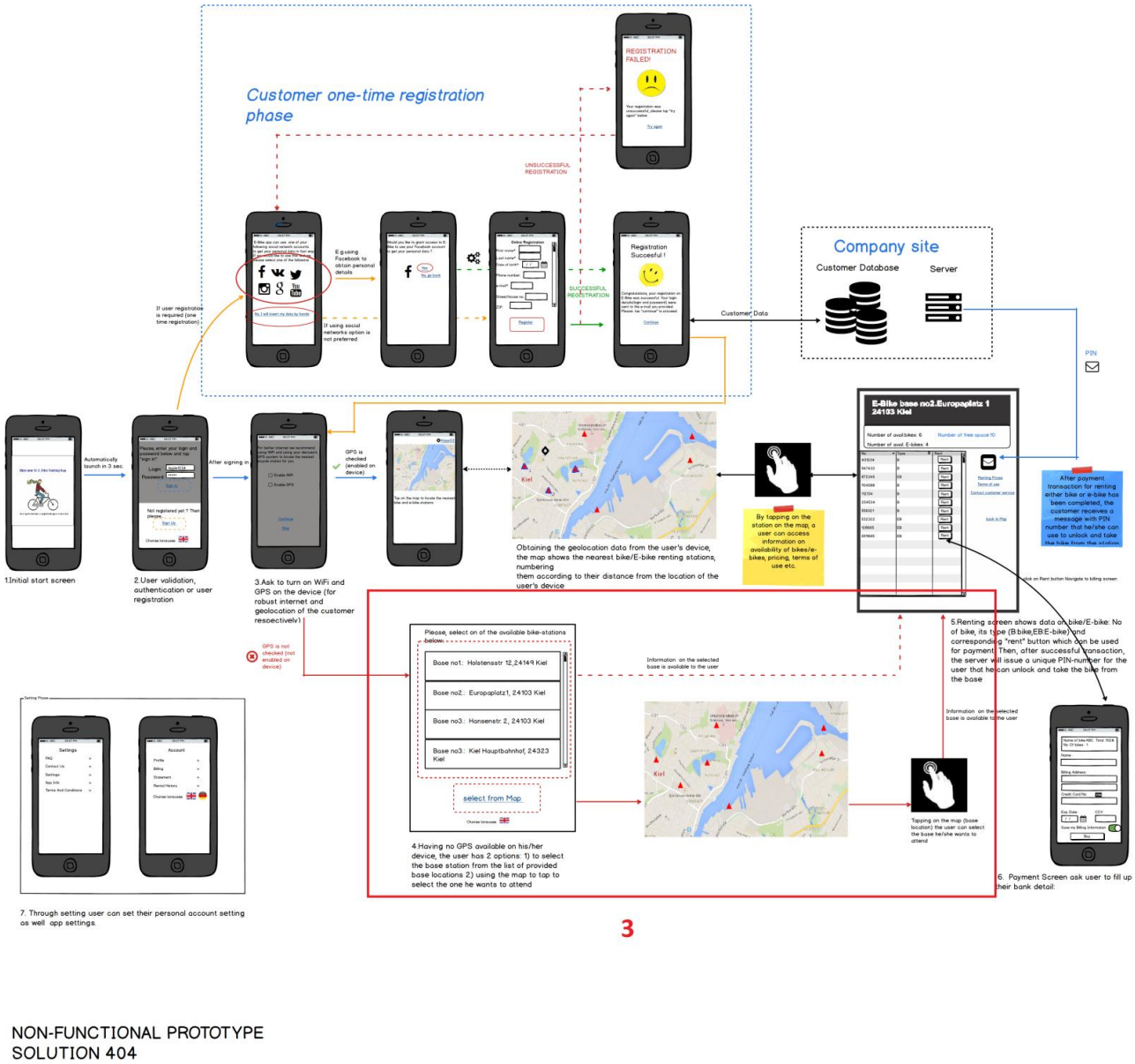


Figure 24.Non-Functional Prototype v.1.0.Offline operation

The Offline mode allows the user to select and rent a bike from a base location of his preference. The only drawback of the Offline mode is that the user can not benefit from the functionality which identifies the user device's location on the map provides distant based indexes on the base location markers.

A detailed view of the Offline mode of the App is given below in the Figure 25.

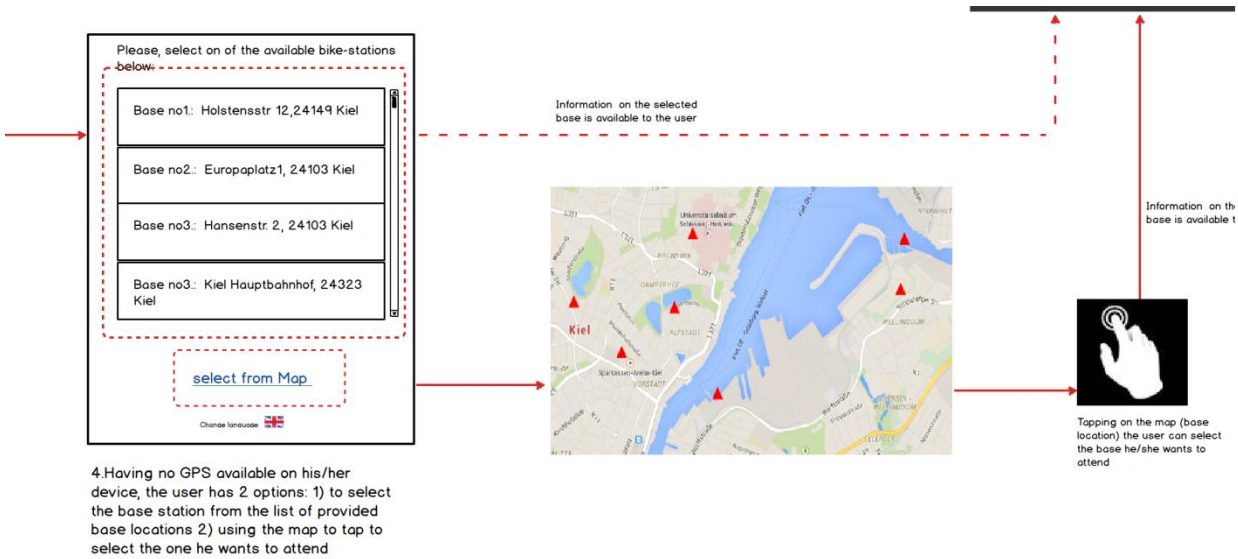


Figure 25.Non-Functional Prototype v.1.0.Offline operation detailed view

As shown in Figure 25, during the Offline mode, the user can not benefit from the Map with location marker identified by the device GPS. However, in the Offline mode the user sees a list menu with detailed information on available bike renting locations spread across Kiel.

The user can either select a base location from the list of provided locations or has an option “Select from Map” to tap on the Map and select a location.

As selection of the desired location was made the user, he/she will see the screen for renting bikes/e-bikes, absolutely identical to the one for renting in online mode.

Payment

As per user's requirements from the user stories, most users preferred to use online payment as main means for paying for renting the bike. Therefore, the App for renting bikes and e-bikes has functionality to make online payments. Below, in Figure 26, it is given prototype mockup for online payment.

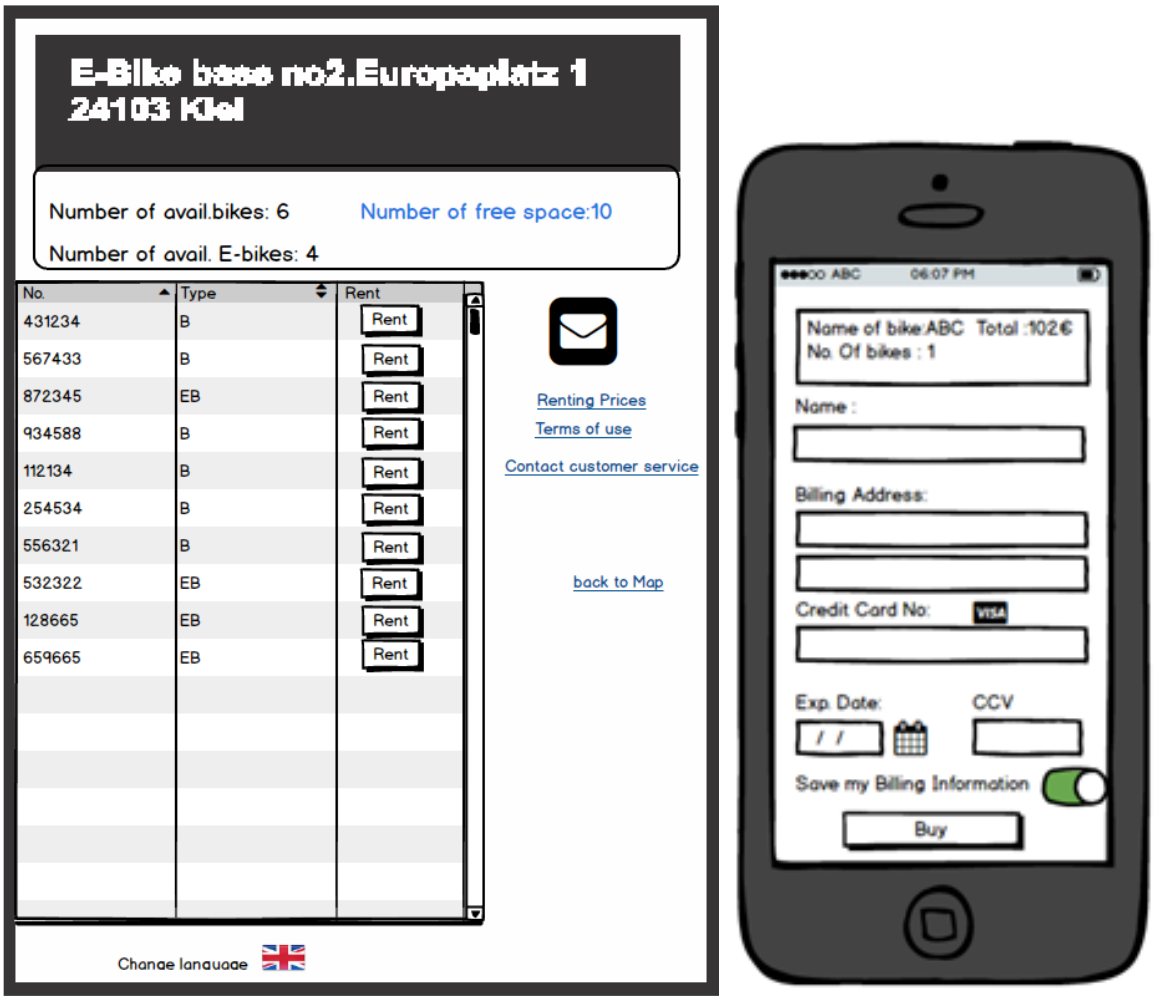


Figure 26.Non-Functional Prototype v.1.0.Online Payment

As it can be seen from the Figure 26., once the user has selected the desired base station which he wants to attend, he sees the screen with information on all bike and e-bikes available at this particular station (bike no.(ID), type (either bike (B) or E-bike(EB))). The user also sees the name of the station with its address. This screen as well provides the information on number of currently available bikes, number of

available E-bikes, as well the number of free spaces available to where the user can return his/her bike or E-bike after using.

After the user has chosen the desired bike/E-bike, and tapped “Rent” button, the screen with payment information appears. Here the user sees the bike/E-bike he/she has chosen, cost, and is requested to input online payment card information.

When the user has entered correct payment information, he can choose to either save or not the online payment card information on his device.

If the online transaction was successful, the user will receive a message with PIN number, which he/she can use to take the bike from the station. The incoming PIN number appears in the personal message box of the user.

Settings

An each user profile is provided with Settings menu [Figure 27.].

Figure 27.Non-Functional Prototype v.1.0.Settings

The Settings menu allow user to read FAQ's, Contact the Freedom Bikes, Change settings, read App Info (see the release version) . The Account settings allow user to manage the user Profile, manage the Billings, Statement and view the Rental History.

6.6 Task 6 – Interview on the non-functional Prototype V.1.0

In agile software development feedback plays an important role. I is widely known how feedback supports dealing with changing requirements. Therefore, when the app prototype was ready, it was shown to the Personas in order to obtain their opinion regarding in what degree the mock-up prototype meets their expectation and requirements.

In total the time given to fulfil this task (24h X 7d = 168h).

All of the Personas, who were interviewed in the second interview were shown the self-explanatory prototype mock-up, which was designed well enough to be understood by individuals away from IT (i.e. no coding and no UML).

The Personas expressed their full satisfaction with the functionalities and the user interfaces. All of the interviewed personas said that the App prototype satisfies their user requirements.

However, one of the Personas (Tobias Kampfe) in his interview said that according to his experience of being abroad in Sweden, he had a problem with finding back his bike in densely bike settled area. It took him a while before he could identify his bike, because all the bikes look similar once they are stowed in line.

Therefore, considering the users' previous experience, as well as that this problem could be identical to Kiel, since in Kiel there is high interest among population to using bikes, it was proposed a solution in the Prototype V.2.0 in Agile manner.

6.7 Task 7 – Non-Functional Prototype for Freedom Bikes app V.2.0

Considering the new user requirement, our user stories was altered with a new user story.

This new user requirement was met in the Prototype for Freedom Bikes V.2.0.

In Figure 28. It is shown the re-designed prototype V.2.0.

Figure 28.Non-Functional Prototype v.2.0.

In version 2.0 of the Prototype, it was added a new Module which is called Bike Locator. The bike locator module is incorporated into the backend of the system, therefore making its security guarded from the backend. The detailed view of the Bike Locator Module is given below in the Figure 29.

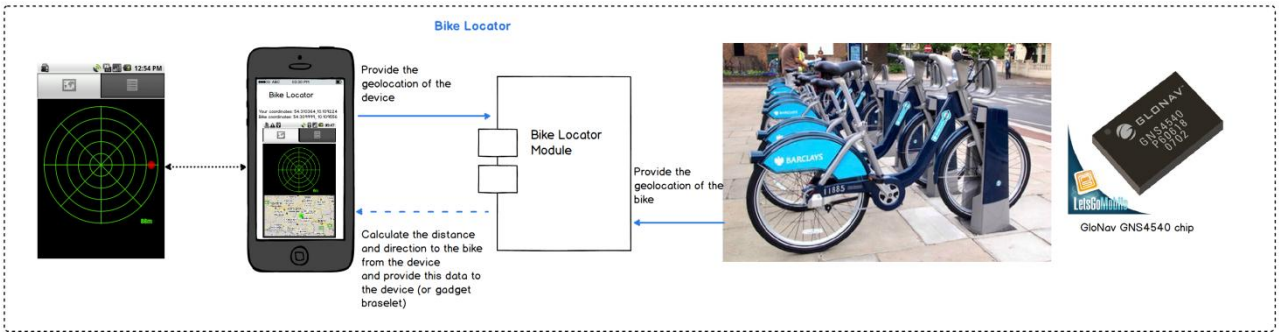


Figure 29.Non-Functional Prototype v.2.0.Bike Locator

The Bike Locator Module is the main module to provide Geolocation services, so that the distance between the user's device and the bike/E-bike between him/her could be calculated. The module acts in the following way: the user device sends GPS based data in the form of longitude and latitude to the central Bike Locator Module. The similar data is sent from the bike/E-bike that the user has rented (since PIN for transaction as well as the bike/E-bike numbers are in relational database, finding out which bike/E-bike to request this data is not a problem).

The algorithm in the Bike Locator module analyses the data provided from the both sides, and produces high precision and accuracy direction as well as distance to which the user needs to take in order to reach his/her bike/E-bike.

The Geo-data provided from the user's device is based on GPS data of the device. However, the Geo-data provided by the bike/E-bike is based on high-accuracy GloNav GNS4540 chip.

Here are some of the features of Glonav GNS4540:

- Best-in-class acquisition and tracking sensitivity, TTFF and accuracy: 157 dBm acquisition / 159 dBm tracking sensitivity in A-GPS and autonomous hot start modes
- Optimized for both cellular handset and portable consumer device integration
- DynaTrak advanced tracking and multipath algorithms for robust low dropout tracking in indoor and outdoor, very low signal and rapid signal transition environments
- Hosted architecture optimized for cost and performance
- Supports GSM, WCDMA and CDMA control plane A-GPS assistance data standards
- Supports SUPL user plane A-GPS assistance data
- Exceeds 3GPP and TIA performance requirements
- Ultra-low power consumption: 30mW in 1Hz tracking mode
- Supports all common GPS and cellular handset reference frequencies (10 – 50MHz)
- Very low external component count and PCB size
- Supports ultra-low cost PCB dipole antennas
- UART and SPI host interfaces
- Operating temperature range -40C/+85C

The chip could be based on the motherboard that might be designed to charge from a battery charging as the user is rotating the pedals of the bike/E-bike.

6.7.2. Non-functional software prototype:

Considering all the requirements from customers for Mobile App we have generated the software for the Freedom bike company only. [Figure 30.] This will be very helpful to handle and manage our bikes within all the base stations.

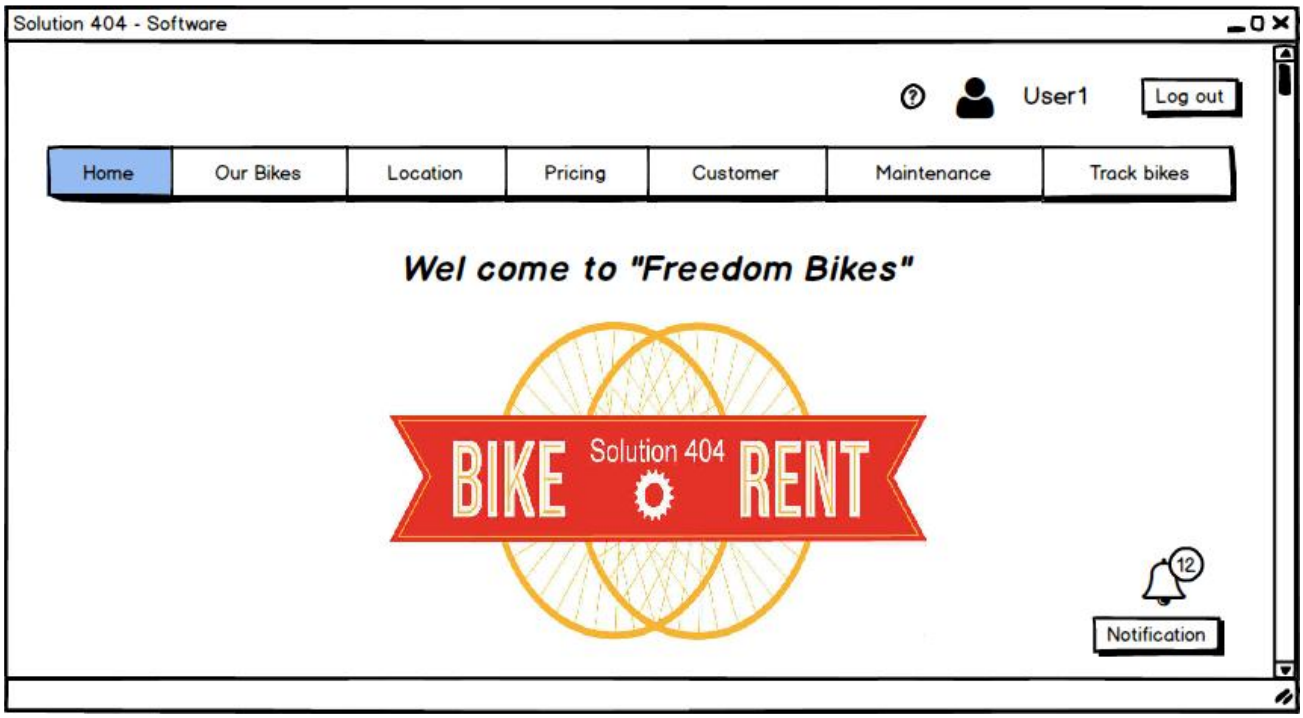


Figure 30. Home page of software prototype

This image describes about the home page look of software window. This software has access of only Freedom bike's employee. This software will be used to handle and manage the bikes to/from customer, maintenance of bikes, details of all base stations and it will notify the employee the latest bike delivery to customer and return date of bikes from customer.

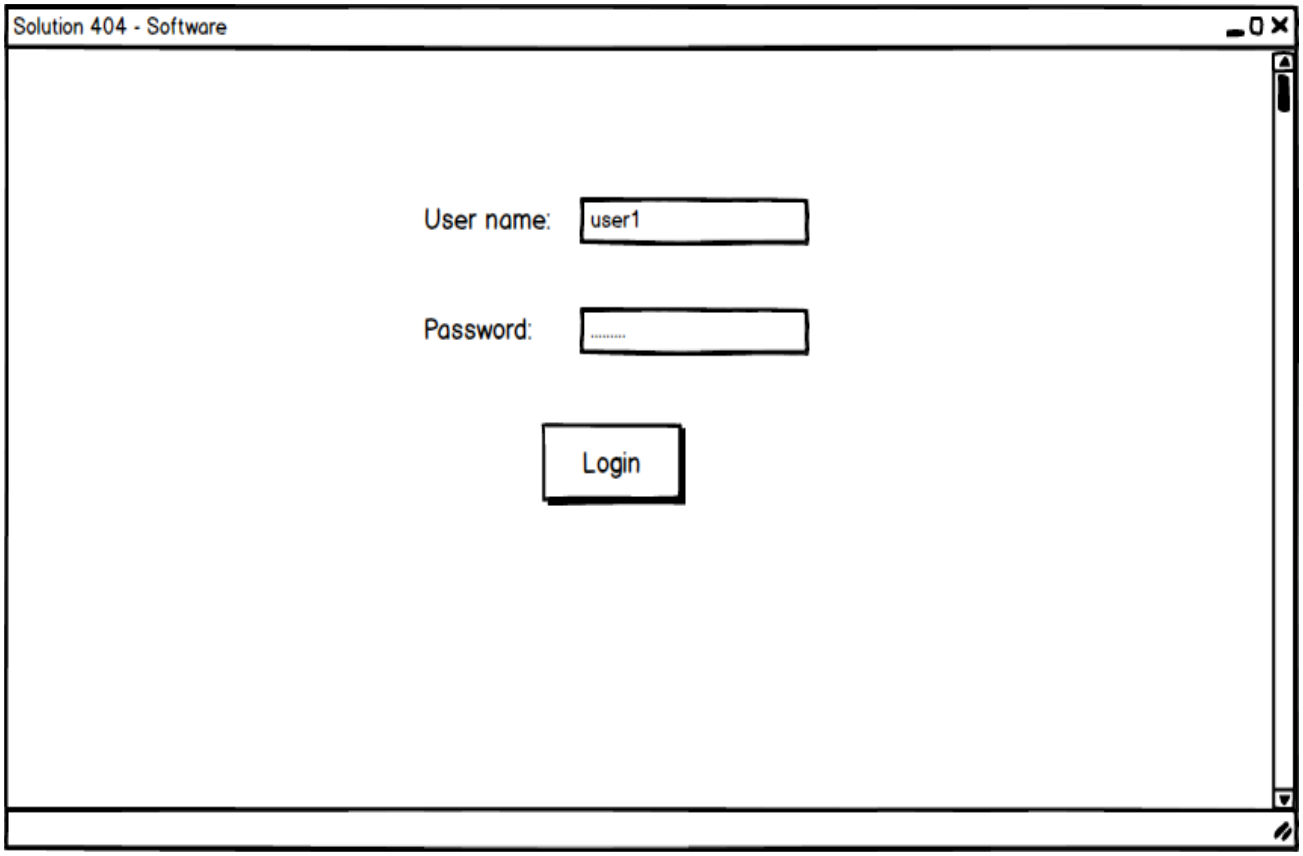


Figure 31. Log in window for user

This window is for log in for user of company. [Figure 31.] All employees will be given their user id an passwords accordingly and only they can access this software. After the complete set up this page will be open to ask user name and passwords. After log into this page user will see the home window of software as below.

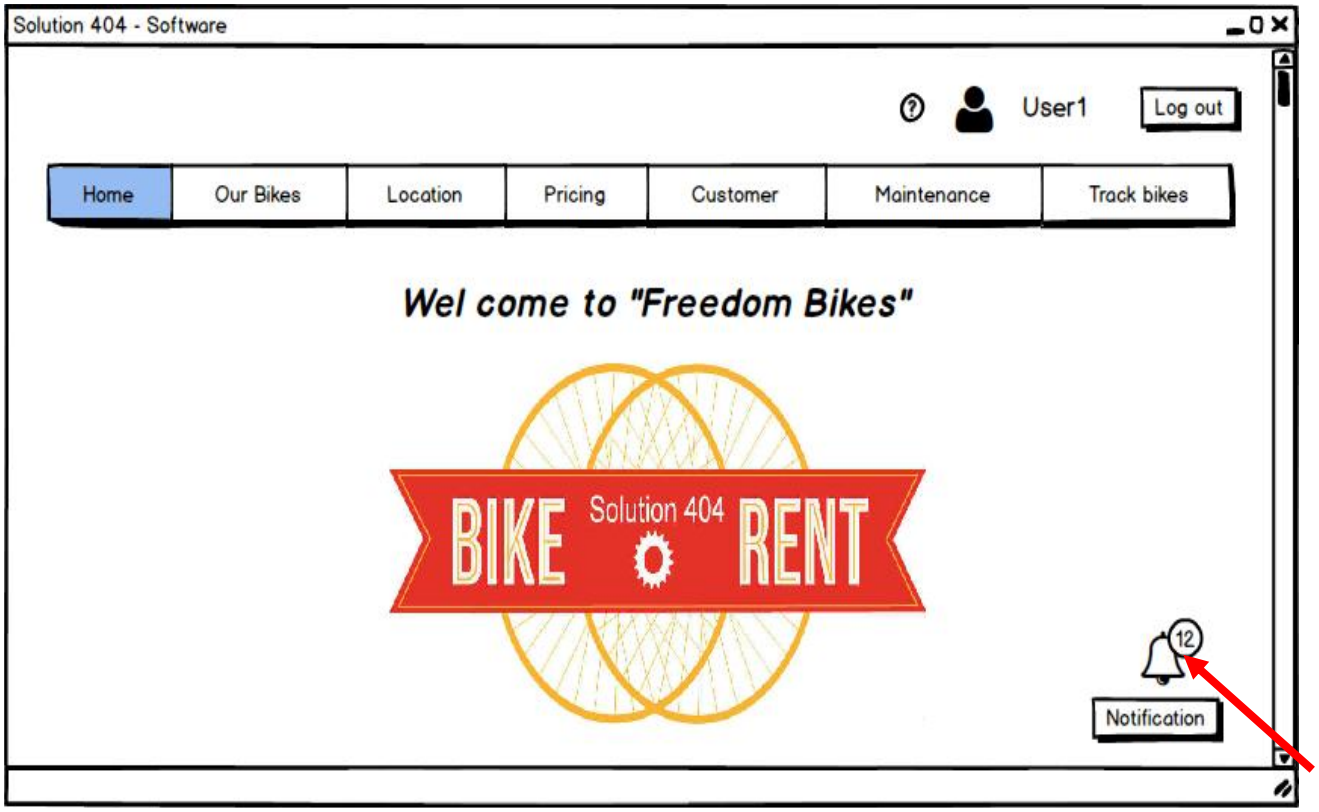


Figure 32. Home page

In the right upper side there is the user name who is logged in and will show the log out button. [Figure 32.] In task menu there are 7 window options:

Home, Our bikes, Location, Pricing, Customer, Maintenance, Track bikes

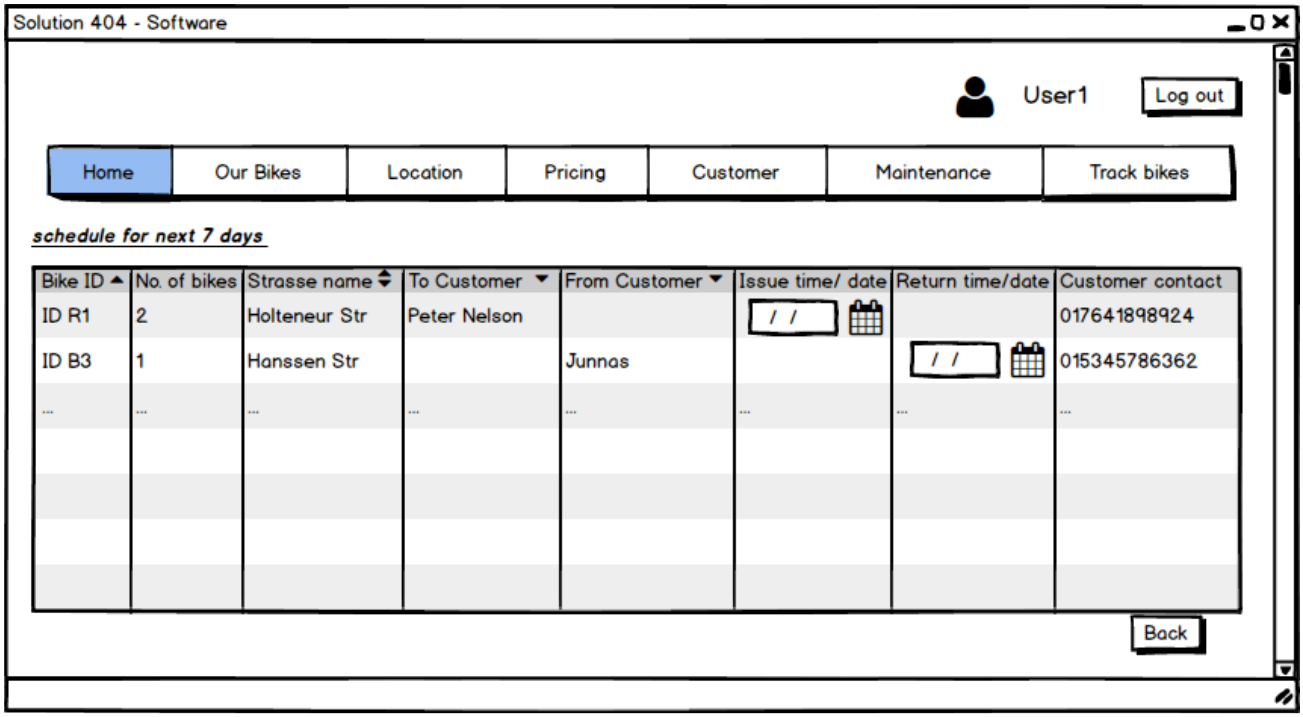


Figure 33: notification window

In home window [Figure 33.], there is the Notification button, that will give reminder of upcoming delivery of bikes and other important task with total number of notifications. This provides the facility of up to date feature. This notifies any chat message or SMS from customer, so operator can be in contact direct with the customer.

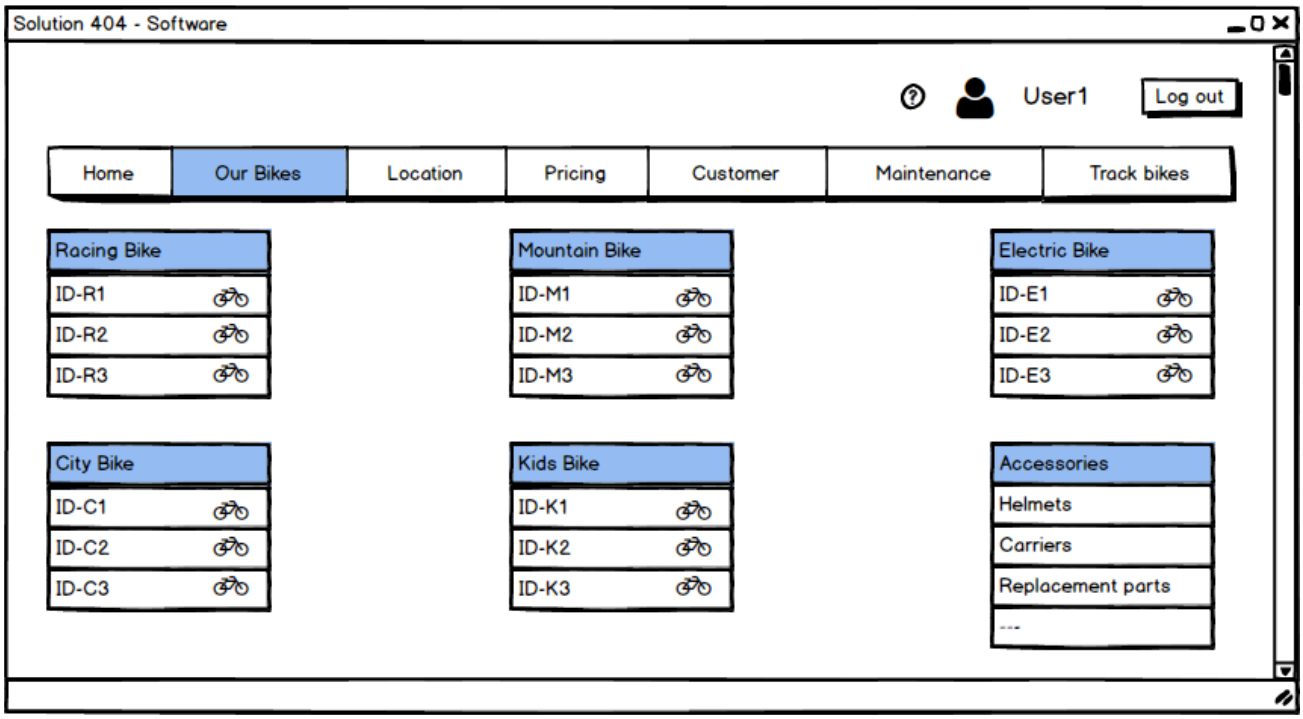


Figure 34: Window of type of bikes with their rent charges

This is wndow describes all the details of bike, the Freedom Bike offers with its rent charges. It gets updated from the server direct as product owner inroduces or changes anything in bike pricing and new bikes. Here all bikes have their id names. For example if bike is racing bike and that too of different style, then the bike will get id as ID-R1/R2/R3.... and so on. Here “R” stands for racing. So accroding to the list in image, “M”- mountain bike, “E”-electric/e-bike, “C”-city bike, “K”- kids bike etc. [Figure 34.]This includes the accessories list also with total accessories in our all base stations. Each accessory is also has its id. This window will only show how many type of bikes we have in our base stations overall.

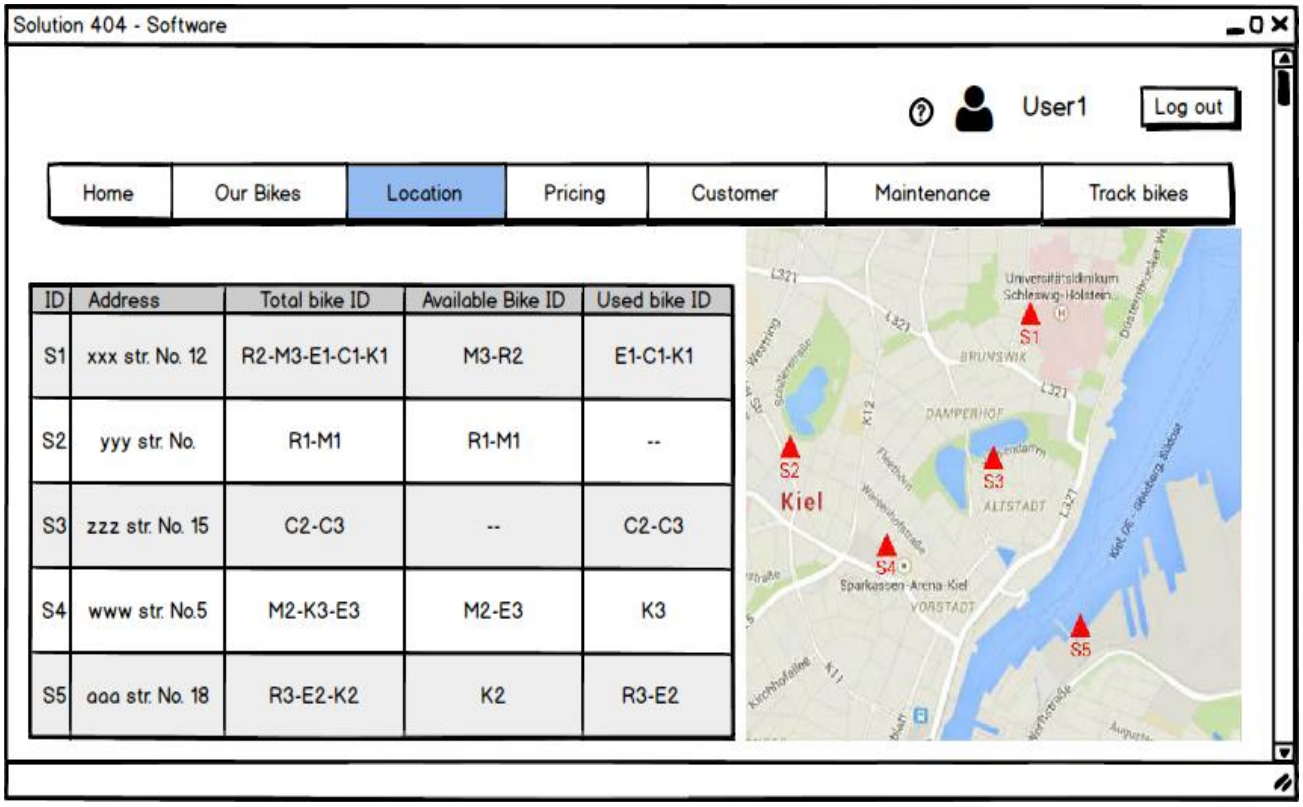


Figure 35: location window of software

This window shows the map, where we have our base stations in different areas of Kiel along with detailed list of bikes. [Figure 35.] This table will show which base station has which bikes and how many bikes from them is on rent and available. So the operator can come to know about availability of particular bike if any requirement or emergency comes.

This table will provide the contact number and address of particular base station so operator can direct contact them for any inquiry.

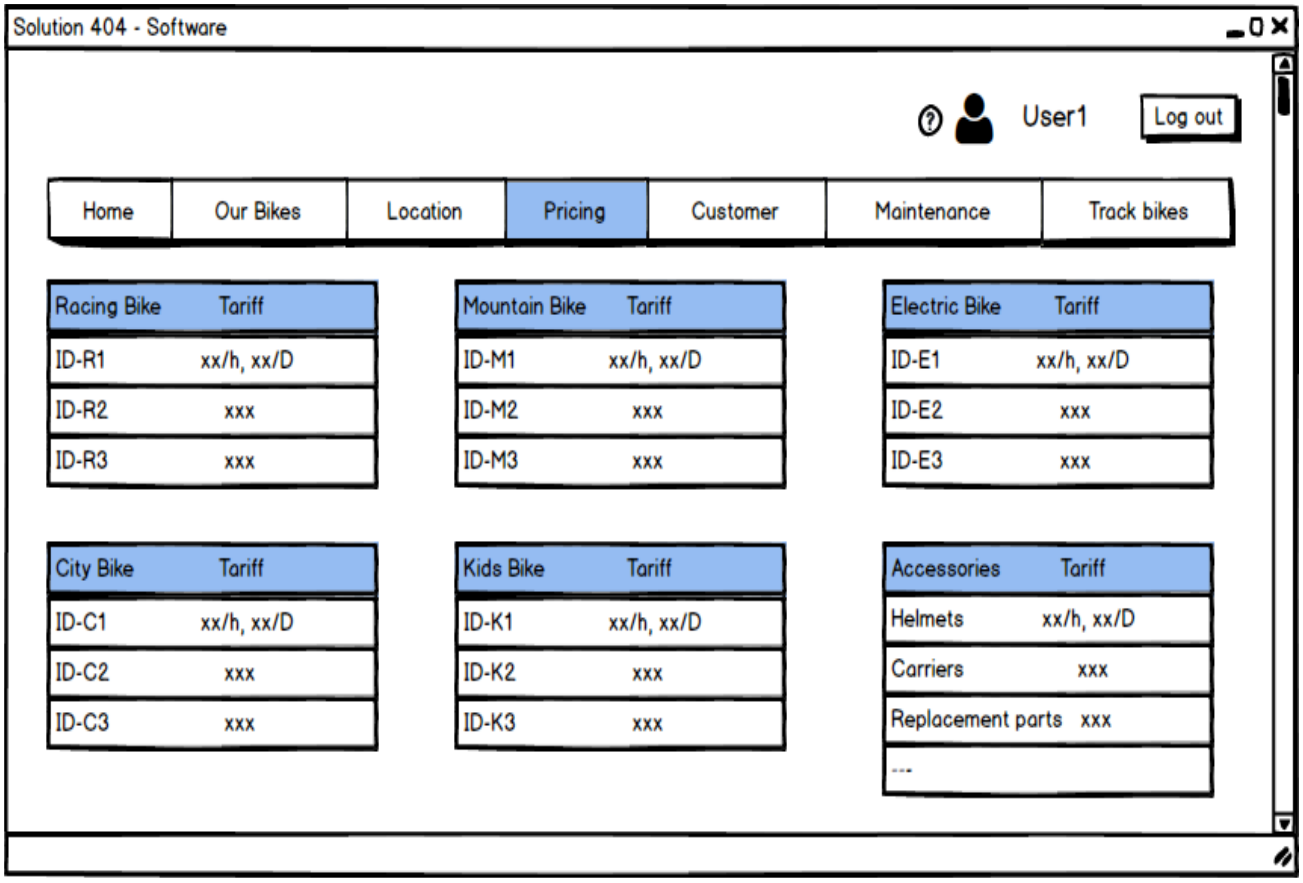


Figure 36: pricing window of all bikes

Our Freedom bike provides 3 types of renting, hour based, day based and energy based for e-bikes. This window is very much similar to “Our bike” window, the only change is the price range of all bikes. Here the ranges got updated from the server. [Figure 36.]

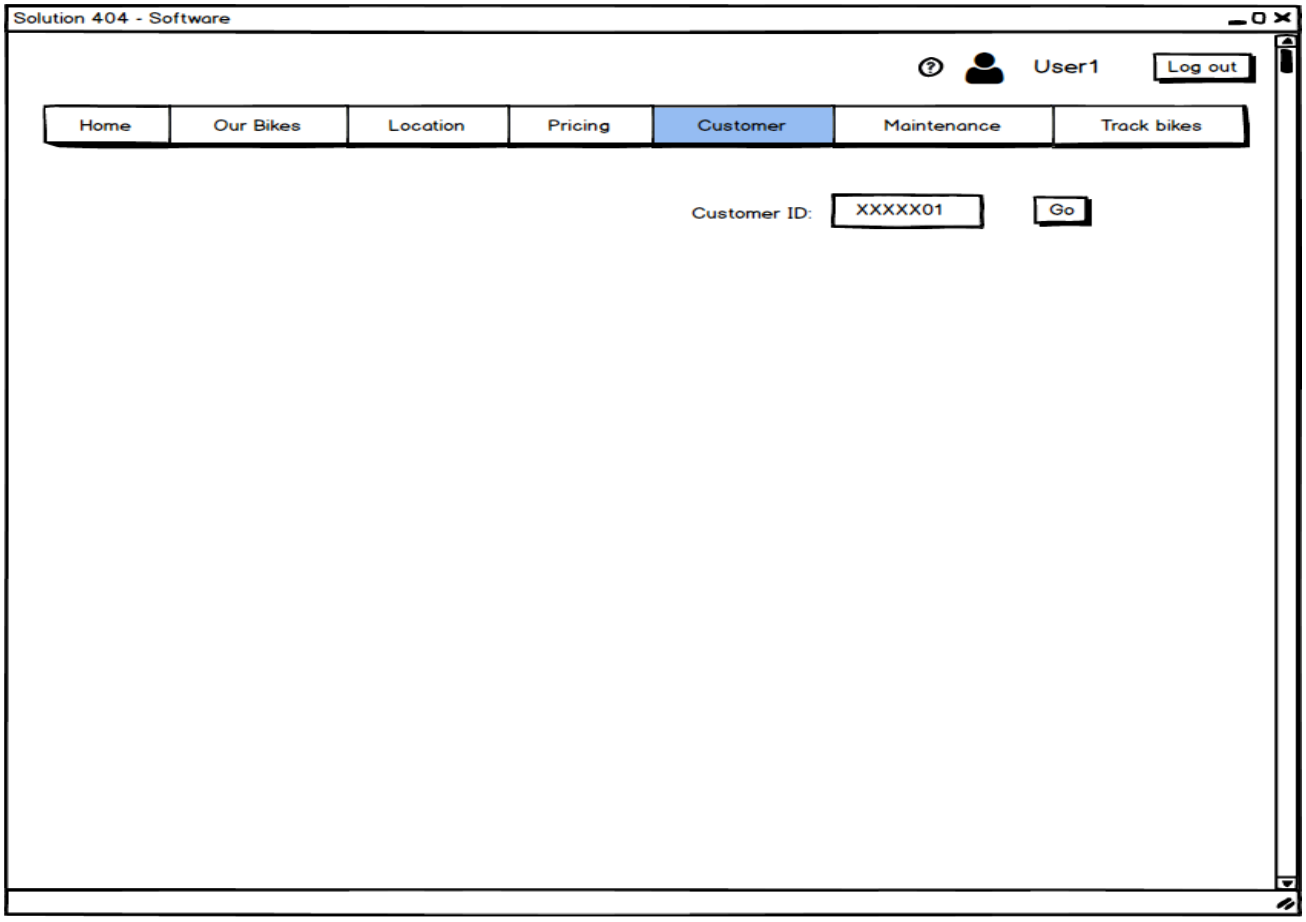


Figure 37. Customer window 1

This window is the most important function of our software as it connects to the particular customer. All customers will get their id number so management will be easier for company. In this window initially we have to write the customer id to get his/her details. [Figure 37.]

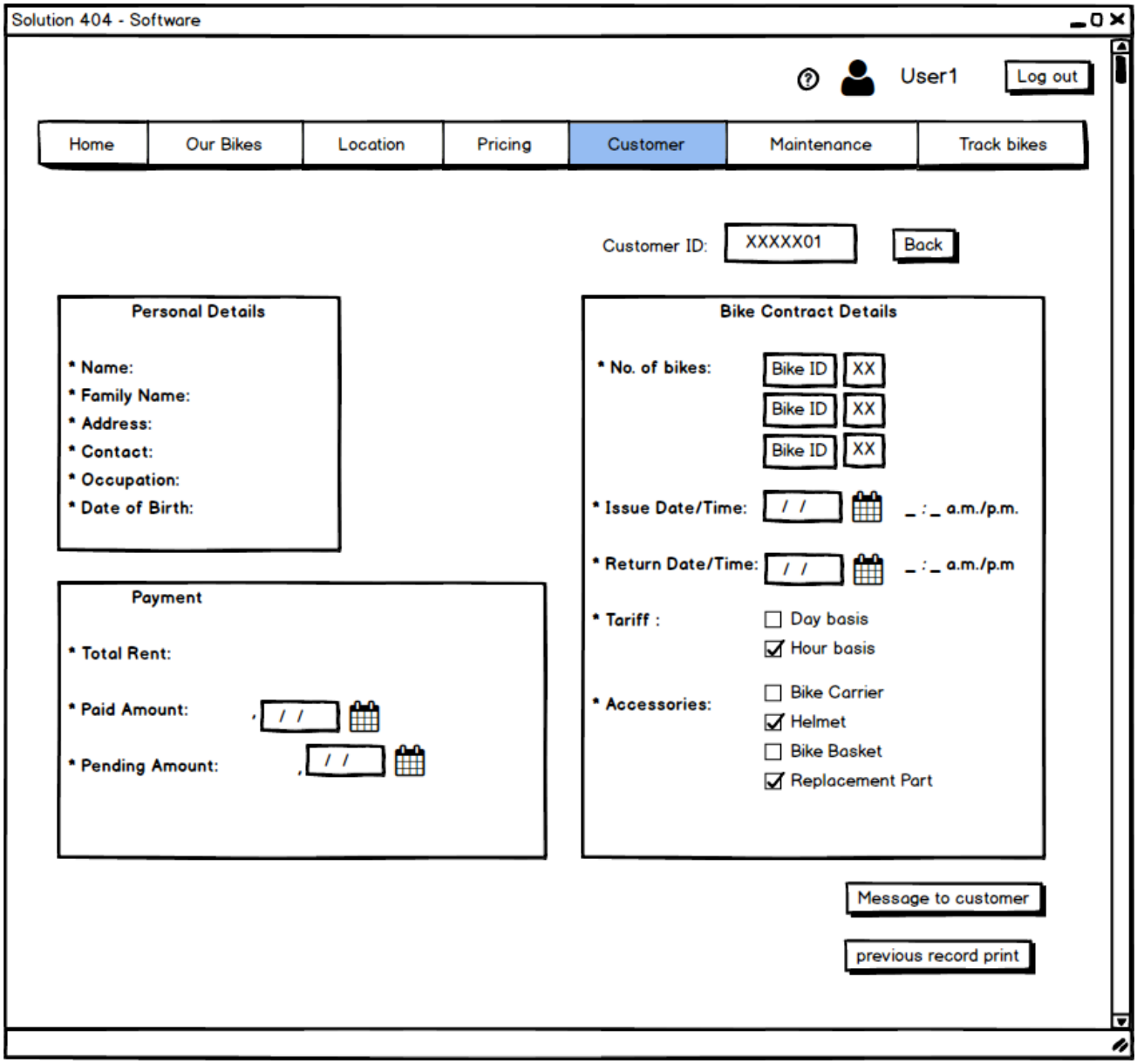


Figure 38. Customer window 2

After entering the customer id, operator will get the new window of that customer's detail. Here he will get details like; customer's personal details like name, address, contact number, occupation, date of birth etc. These details may be come from their any social network as our mobile app gives option to them to get theri personal data from there. The second block is bike details which they have taken on rent. This block will show the information about the date of issue and return date, no of total bikes with bike ids, tariff which customer has chosen (day/hour/energy based) and any accessory

if they have purchsed or rent. The third block is of payment from the customer. This shows the total amount of rent if custmer has predecided his/her use and also paid amount and pending amount. These all details will be shown with the time and date. [Figure 38.]

This window also facilitates the online chat and SMS to customer directly. The contact number will be preloaded or saved in the server so it will send the message from the operator directly on the phone of customer. So customer can be in contact online and offline as well. There is the button called “message to customer”, if it is clicked then it will connect the operator to the chat window as shown below.

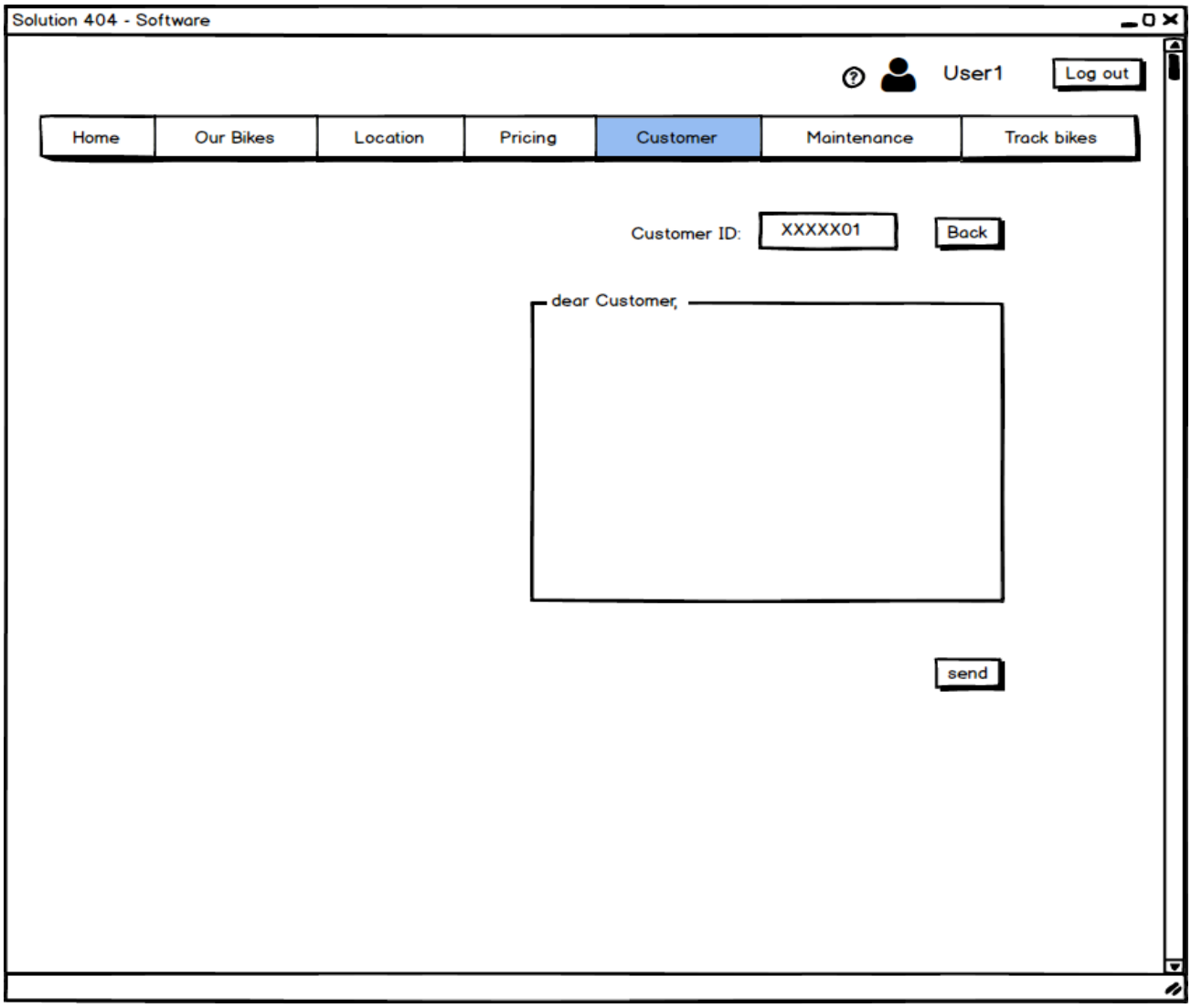


Figure 39: customer chat window 3

Here the small chat window is available. [Figure 39.] After writing a message, operator will send it to customer, and there customer will receive SMS on his phone.

There is one more option for customer. That was “previous record”, which gives previous or past record of customer. If the older customer again rent a bike then his/her previous documents will be available. So operator can look into it for any need.

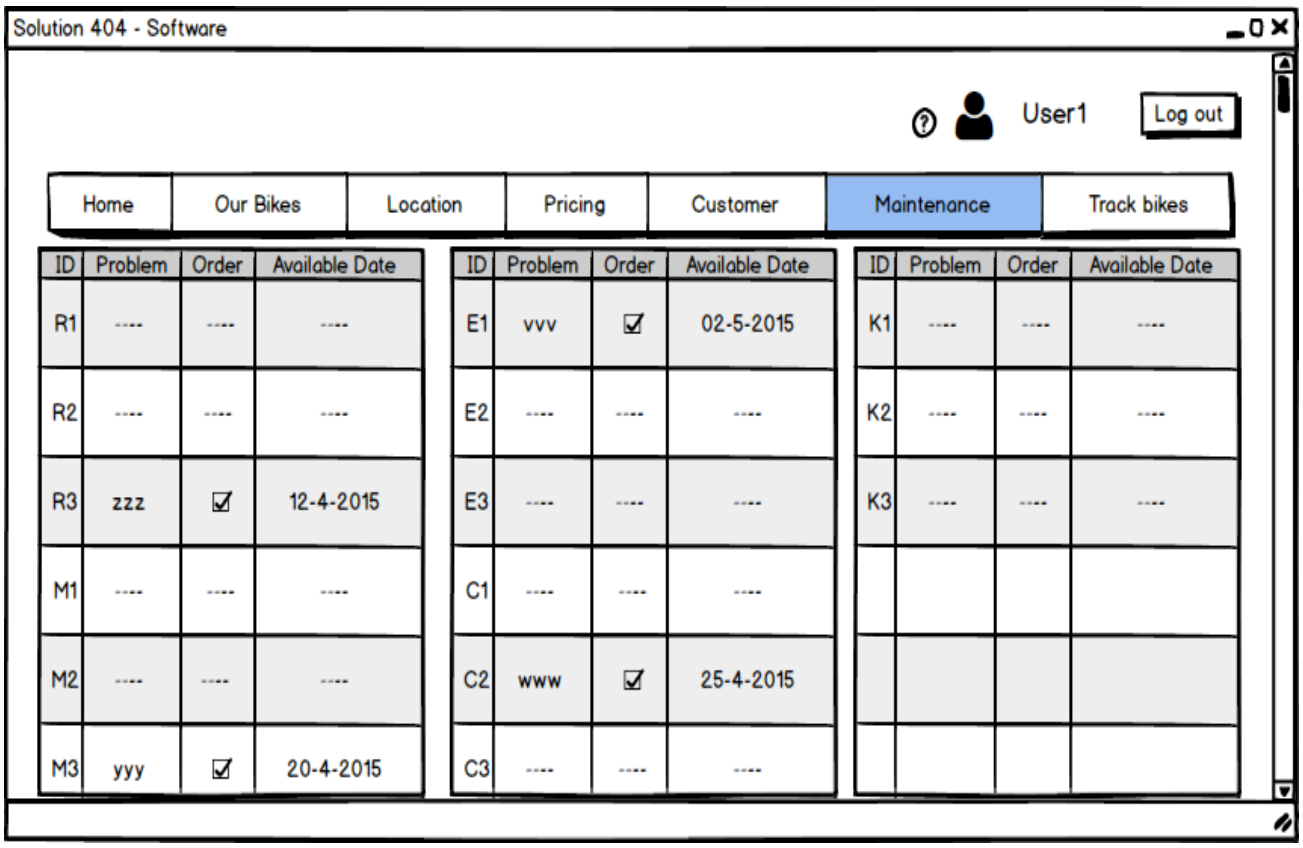


Figure 40: Maintenance window

This is the window for bike maintenance. This shows the list of bikes which are under maintenance or repairing. Operator can easily figure out that which bike is in maintenance and what is the problem of particular bike and the order number of it and on which day it will be available again. So he can manage those bike for customers and make them ready for the rent and sale. [Figure 40.]

From this he can even do inquiry in maintenance department for the early repairment, and the solution if any urgent requirement comes.

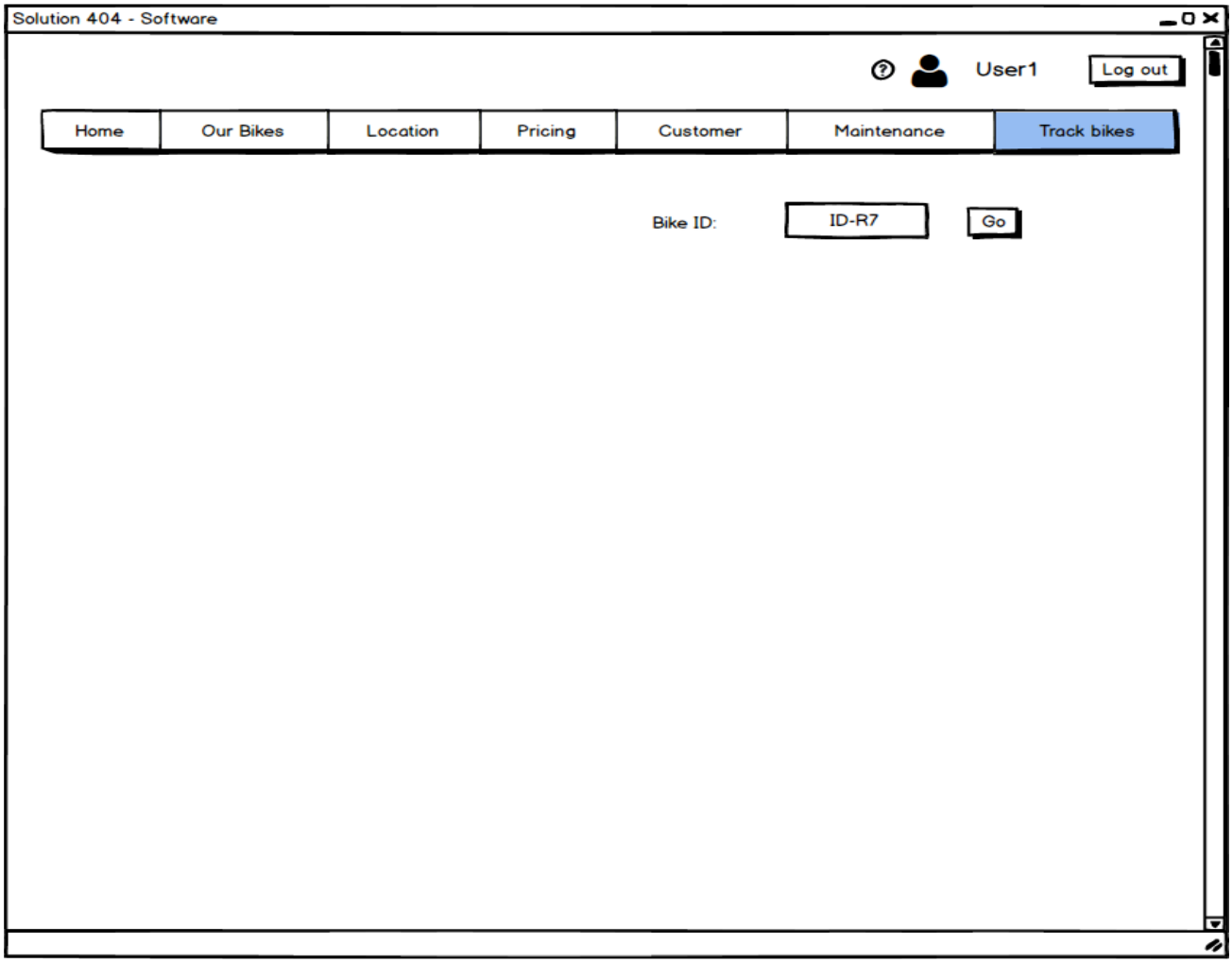


Figure 41: Track bike window 1

For any need or requirement, the operator can track bikes whether it is on rent or available in warehouse or in any base station. He just has to write the bike id for tracking.

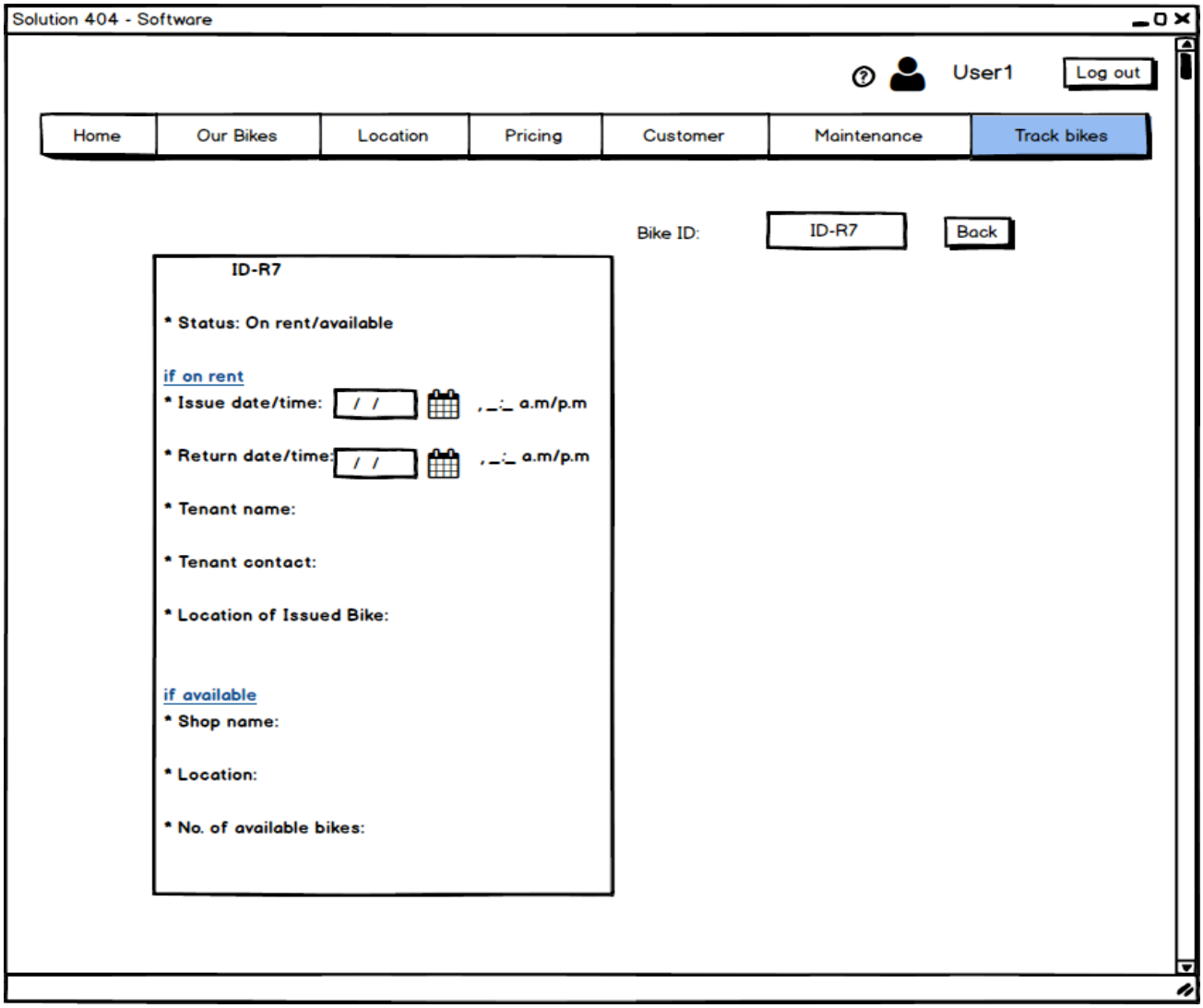


Figure 42. Track window 2

After writing the id number of bike, this new window will be opened which will show the actual detail of that bike like; status (on rent/available). [Figure 42.]

If bike is on rent, then the tenant name and his contact and the issue date & return date of that bike. And if bike is available, then in which shop or base station the bike is available and the location of it and how many bikes of that category is available.

By using this window operartor can easily manage the bikes within the base stations and customers.

List of bike rates:

Figure 43.racing and mountain bike rent price

Figure 44: E-bike rent price

Figure 45: kids bike rent price

Figure 46: accessories renting price

6.8 TASK 8 and 6.9 TASK9 [Product backlog & Sprint Backlog]:

SCRUM

There has been a dramatic change within last one or two decades in how software product is developed and deployed. Nowadays software becomes crucial and essential part of our society. There is no business firm or aspect of our life is not affected by software. Business sector, politics, and education, entertainment, and infrastructure activities all area operated and governed by software. Before a decade ago most software build and produced by traditional software development process called Waterfall Model. Waterfall model is a sequential approach towards software development in which output of one phase act as an input for next phase sequentially. In figure 47, shows the steps of waterfall model.

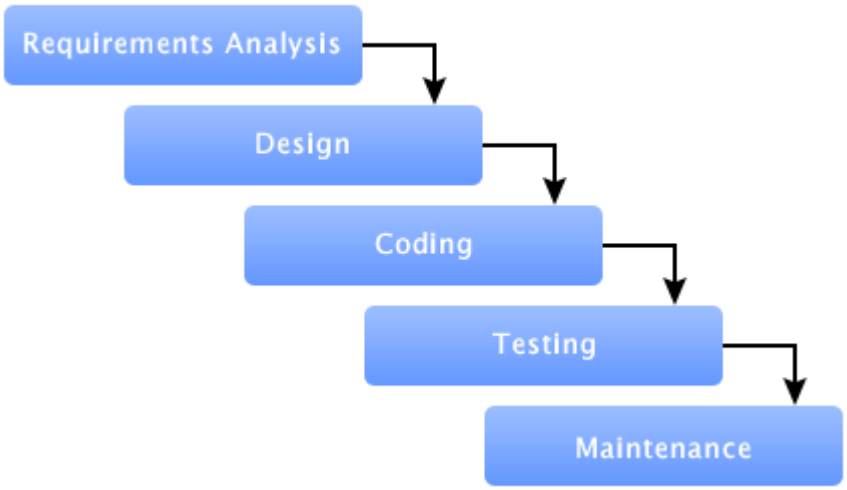


Figure 47. Waterfall Model

From figure 47, we can easily understand and utilization of software development process in waterfall approach. In this approach each phase of life cycle must be completed before start of a new phase. With the low customer action involve during development and after only product is developed then it delivers to the customer. For dynamic requirement change of customer fails the traditional approach of waterfall model and thus there could need to develop another development approach compatible with dynamic change requirement with best outcomes.

Scrum is an iterative and incremental agile software development framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal". It is software development process with new roles in team with variety of new challenges. Scrum methodology is developed by **Ken Schwaber** and **Jeff Sutherland**. According to founder of scrum

“A scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.”

The scrum framework consists of scrum teams and their associated roles, artifacts, events and rules. Each part of the frame work deliver specific outcomes and its essential to scrum success and usage.

6.10. Roles in Scrum

Scrum roles consist of several individual architects like Product Owner, Scrum master, scrum team, stockholders and customers. Moreover scrum team is a collection of individual working together to deliver the requested and committed product increments. To work perfectly and efficiently it is crucial for scrum team follow the common goal, same norms, rules and show respect to each other. While developing scrum team there is always measurement of delivering highest possible performance from the beginning. Typical scrum team is about five to nine people .with priority of very high project scrum project is implemented or completed using team of scrum team.

6.10.1 Product Owner

Chandni Jariwala perform role for product owner. She is importantly a project's key stake holder. Part of the product owner responsibilities to have a vision of what she wishes to build and convey that vision to the scrum team. Product owner can be lead user of the system or someone from marketing. She always has in mind with proper understanding of user requirement, market value of the software and future trends of the software being developed. She working closely and synchronously with all the stake holders within organization and Here there are some important points for which as a Product owner she is responsible for

- ✓ Manages the product backlog
- ✓ Prioritizing according to business value
- ✓ Demonstrate solution to the key stake holders who were not present in normal iteration demo
- ✓ Announces releases
- ✓ Ensure product backlog is visible transfer and clear
- ✓ Negotiate priorities of user stories, scope, funding and schedule
- ✓ Provide vision and act as behalf of customer

She builds to product Backlog with prioritize user stories from low level, medium level and high level. She picks up mainly important user requirements and build product backlog with 17 user stories.Figure 48, shows the product backlog with 17 user stories.

ID	AS a/an	I want to..	So that..	Notes	Priority	status
c1	customer	I can get benefit from a system in my app	I can locate my bike or e-bike when needed		Highest	****
c2	customer	I would like to get price range and features to get bike	I would choose bike accordingly.		Medium	****
c3	customer	I would like to get bike for daily and hourly bases	I can manage rent of bike according to my usage.		Highest	****
c4	customer	I want that software should keep my personal data confidential and secure	nobody else can misuse my data		Highest	****
c5	customer	I need online chat system in software	I can communicate directly and immediately and fulfil my requirement.		Highest	****
c6	customer	I like to have facilities like GPS in product	it can guide me for new roots in my journey.		Lowest	****
c7	customer	I want a multi-language supported app for the foreigner	they can operate in their own terminology.		Highest	****
c8	customer	I want Membership and Discount offer options in my smart phone app	I can save my money.		Lowest	****
s1	supplier	I need the software should be flexible enough to provide user information	I can make supply accordingly		Highest	****
s2	supplier	I would like that administrator give me base location and number of bike	I can provide stock accordingly		Medium	****
o1	owner	I like that software should be user friendly	I can reach maximum number of customer		Medium	****
o2	owner	I need that software should have feedback system	I can reached to customer requirements.		Highest	****
o3	owner	I like to make the services reachable easily to customers	my customers will be satisfied and my business will expand.		Medium	****
d1	developer	I need clear and appropriate user requirement	I can develop software accordingly.		Medium	****

d2	developer	the whole team should be aware of the needs of customer exactly	no mistake was done while developing a software, because while developing a software all features would be inter-related		Highest	****
d3	developer	I need to develop new ideas to make the application view better, attractive and useful	it will be easier to get familiar with the software		Lowest	****
d4	development team	we need to be able to contact the supplier, owner	we can question them		Highest	****

Figure 48. Product Backlog

6.10.2 Scrum Master

Chetankumar Dobariya performs role as Scrum master in Freedom bike solution and he is responsible for managing the scrum team and facilities the team in the best way such that team can deliver best outcomes. In another way he acts as a coach for the team and helping the team for the best deliverables. He also responsible for making sure that scrum team lives by values and practices of scrum. He always deals with the problems of the scrum team to make them confident and active towards their goal.

Here are some role points of the scrum master.

- ✓ Responsibilities to remove impediments from the scrum team
- ✓ Ensure the team is working well together ,well synchronized and highly motivated towards task
- ✓ Provide supportive learning environment for all who are in the scrum team
- ✓ Enforcer of rules of scrum
- ✓ Often chairs key meetings and challenges the team to improve
- ✓ Act as a servant leader

His task is to encourage and facilitate the team. He also ensures that scrum team follows all the rules of scrum methodology development. He is not also responsible for removing conflicts and impediments but also gives reward to the team having good

work deliverables and also celebrate success to make the moments memorable for him as well as for entire scrum team.

These are the Impediments raised in our scrum meeting during the product development are

- ✓ My ____ broke and I need a new one today.
- ✓ I need help debugging a problem with _____.
- ✓ I'm struggling to learn _____ and would like to pair with someone on it.
- ✓ I can't get the tech support online for group to call me back.
- ✓ I can't get the ____ group to give me any time and I need to meet with them.

6.10.3 Scrum Team (Khaled, Murad, Abdul Razak)

The scrum team is a cross functional team that is responsible for software development. In our solution the scrum team include Backend developer **(Murad Eminov)** frontend programmers **(Abdul Razak and Amir Doorandish)** and Tester **(Khaled Mosharaff).** The team works together to achieve desired goal for making freedom bike solution successfully.Activities and task of each team members are assigned and synchronized in such a way that the targets associated with a specific sprint are achieved. Murad is responsible for the general structure and design of the software product however Abdul Razak and Amir Doorandish work on programming the software. Bugs and errors are identified by the Khaled within the particular sprint of software development with agile methodology. Scrum team responsible for communicating the status of the project daily to Chetan Dobariya. Scrum team is the one around which task associated with product owner and scrum master are assigned. Team members identify complexity of task and allocating efforts.

6.11. Events in Scrum

There are following meetings in Scrum: Sprint Planning Meeting, Daily Scrum Meeting and Retrospectives.

Figure 49. Scrum meeting

6.11.1 Sprint

Sprint is a set period of time during which specific work has to be completed. Sprint execution time is normally between 1 week and 1 month. Sprint is a basic unit of development in scrum. Product-backlog which is the prioritized list of user requirements in the form of user stories implemented within the sprint. Each sprint contains user stories in the form of task to be completed within the time. “Sprint is basic unit of development in scrum”. It is also **time box** effort.

As a scrum master **Chetankumar Dobariya** developed three releases backlog from the product backlog provide by product owner. Furthermore, he prioritizes the each use stories from low level to maximum level according to requirements.

He divided each release backlog in several sprints for our solution freedom bike. He made three sprints for first release backlog and two sprints for rest of two release backlog accordingly.

He set the execution time for First **Release backlog** for 2 months 8 days shows in **figure 50**, second release backlog for 2 months 8 days shows in **figure 51** and third release backlog for 2 weeks and 2 days shows in **figure 52**.

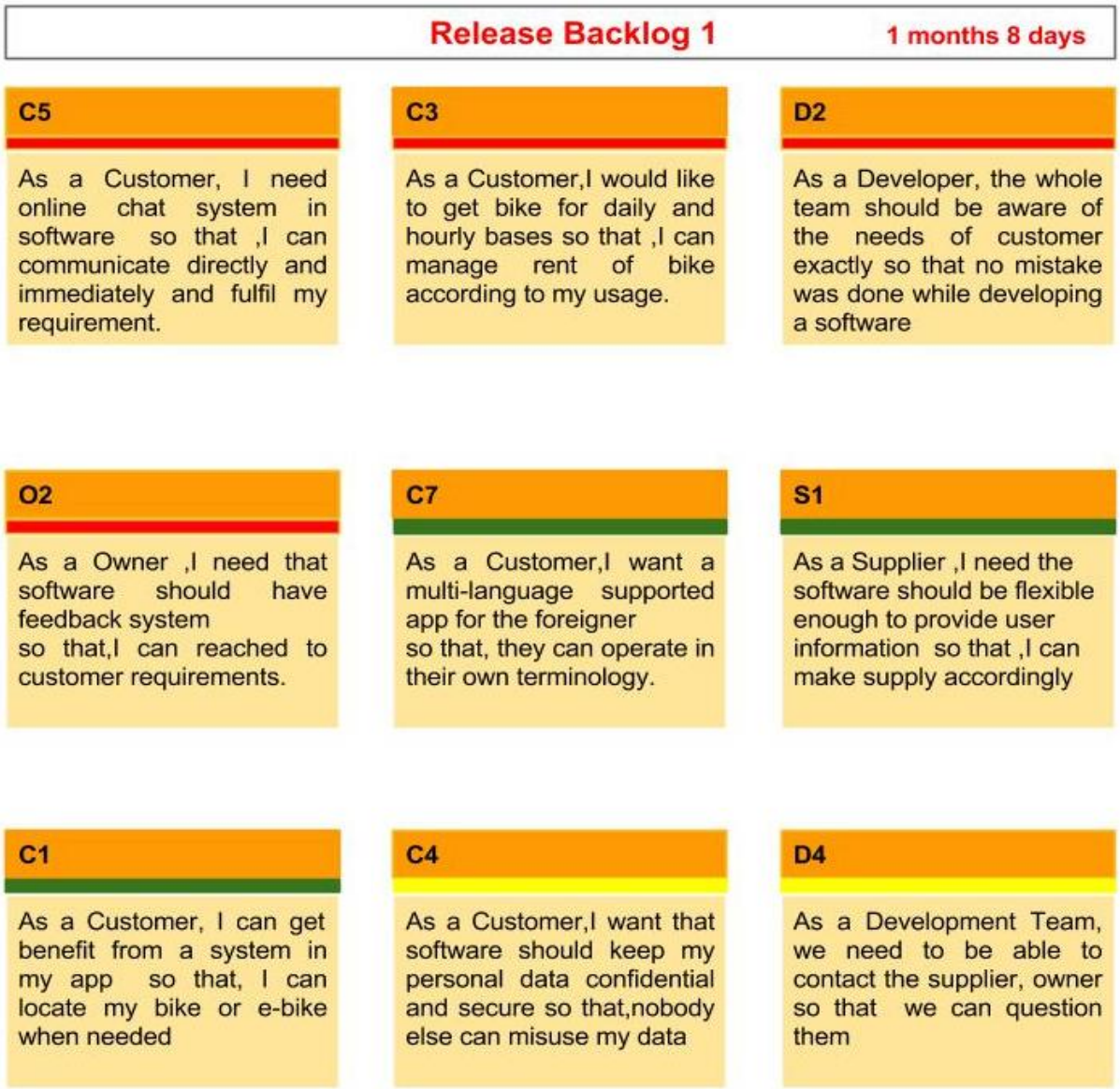


Figure 50. First Release Backlog

Release Backlog 2 **1 Months**

D2

As a Developer, I need clear and appropriate user requirement so that I can develop software accordingly.

O1

As a Owner ,I like that software should be user friendly so that, I can reach maximum number of customer

O3

As a Owner ,I like to make the services reachable easily to customers so that, my customers will be satisfied and my business will expand.

S2

As a Supplier ,I would like that administrator give me base location and number of bike so that ,I can provide stock accordingly

C2

As a Customer, I would like to get price range and features to get bike so that ,I would choose bike accordingly.

Figure 51. Second Release Backlog

Figure 52.Third Release Backlog

6.11.2 Sprint Backlog

A sprint backlog is a list of product backlog items the team commits to delivering.

Each task in a sprint backlog usually estimated. The success of the sprint will later be assessed during the sprint review meeting against each specific item selected from the product backlog.

Chetankumar Dobariya divides the first release backlog into three sprints shows in figure 53. First sprint execution time is 2 weeks 2 days, second sprints have 2 weeks 1 day and third sprints have 1 week 1 day as well. For the second release backlog divide into two sprints 2 weeks for the same and third release backlog divide into two sprints 1 week approximately. Furthermore second release backlog divide into two sprints shown in figure 54 and third release backlog divide into two sprints shown in figure 55.

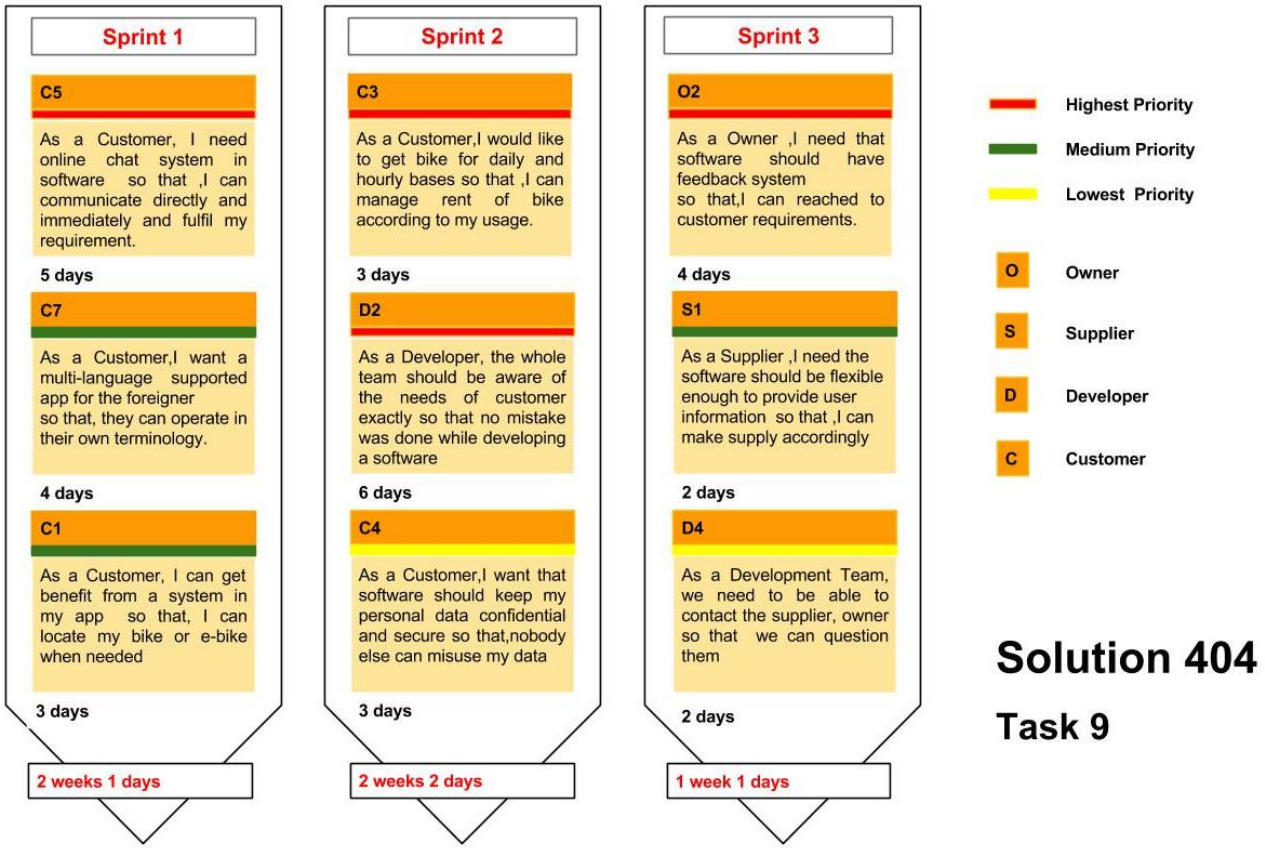
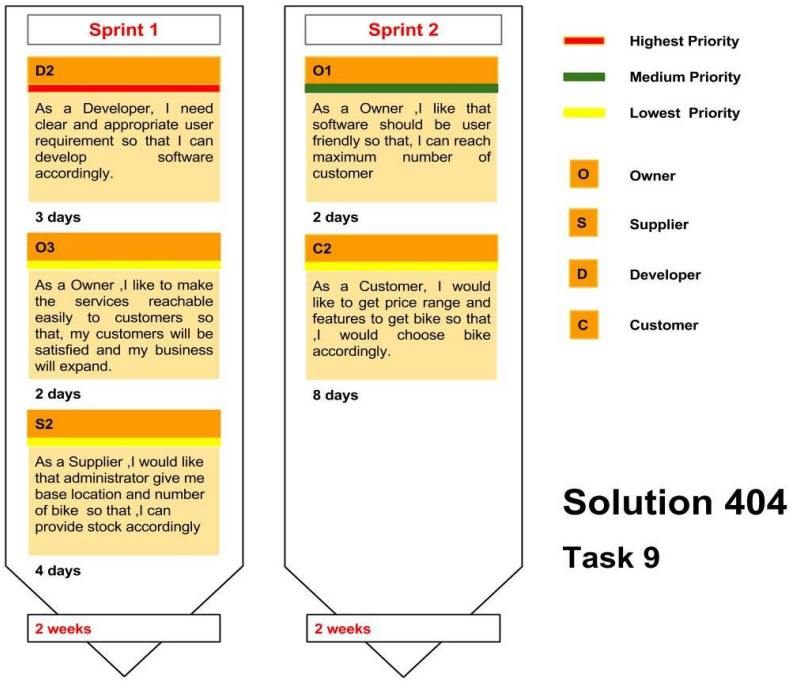


Figure 53. First Sprint Backlog



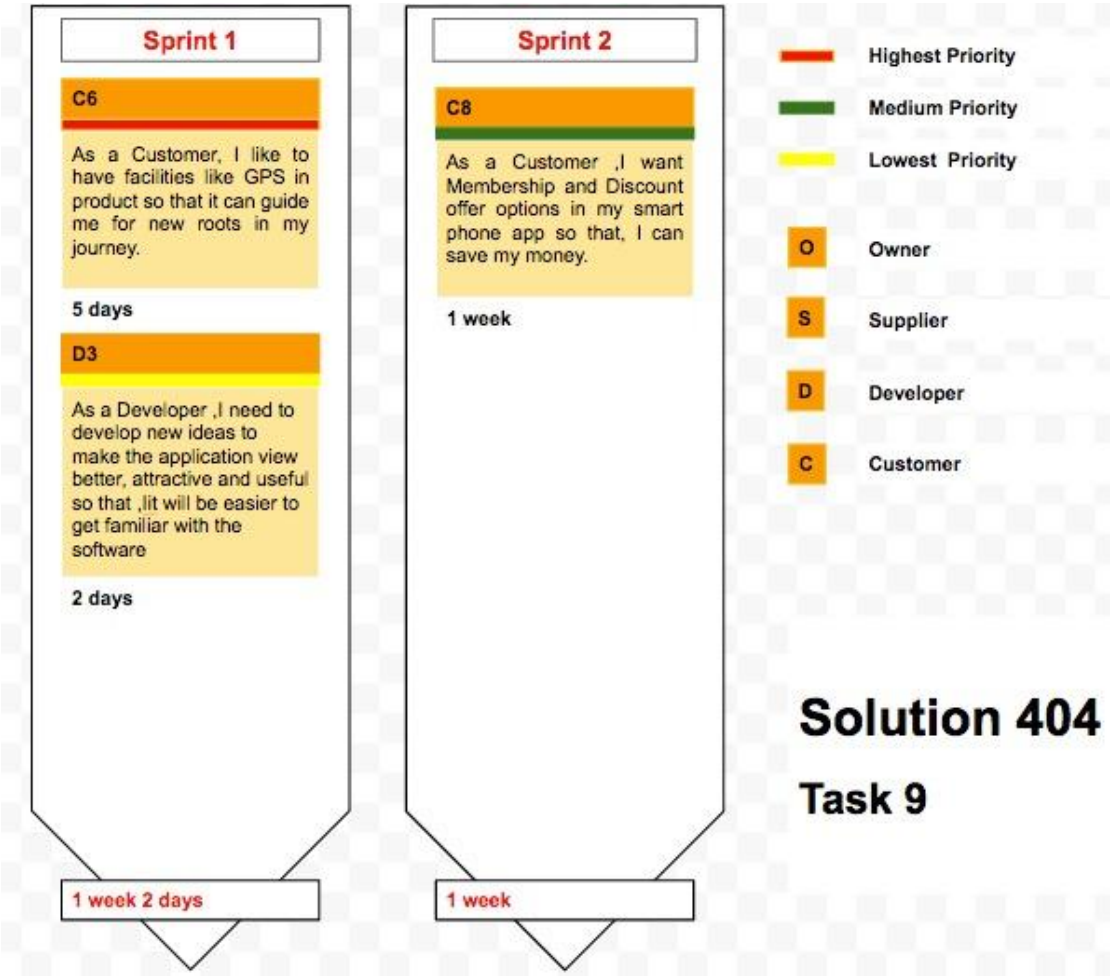


Figure 54. Second Sprint Backlog

Figure 55. Third Sprint Backlog

6.11.3 Burn down Chart

Burn down chart measure progress of development of sprint against a release plan of the sprint .It represents the amount of work remaining against time. Burn down chart is updated after each sprint by scrum master. However according to my perspective view burn down chart is not so much important because there is lot of technologically sound softwares are available in the market which are used to measure remaining task so this burn down chart is not useful for the measurement of remaining task.

For our solutions, we have completed three sprints from first release backlog and we spent 114 hours in 16 days for first sprint shown in figure 56, 110 hours in 15 days for second sprint shown in figure 57 and 52 hours in 6 days for first sprint shown in figure 58 respectively.

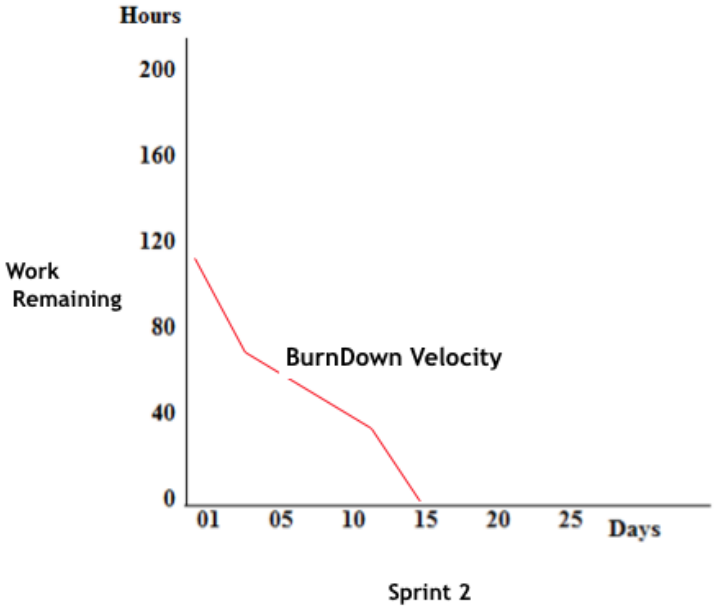


Chart for Sprint 1

Figure 56. Burn Down Chart for Sprint 1

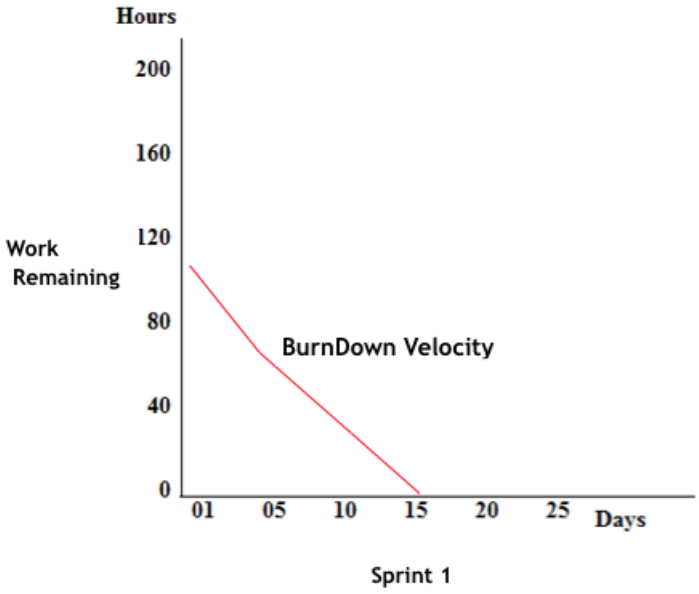


Figure 57. Burn Down Chart for Sprint 2

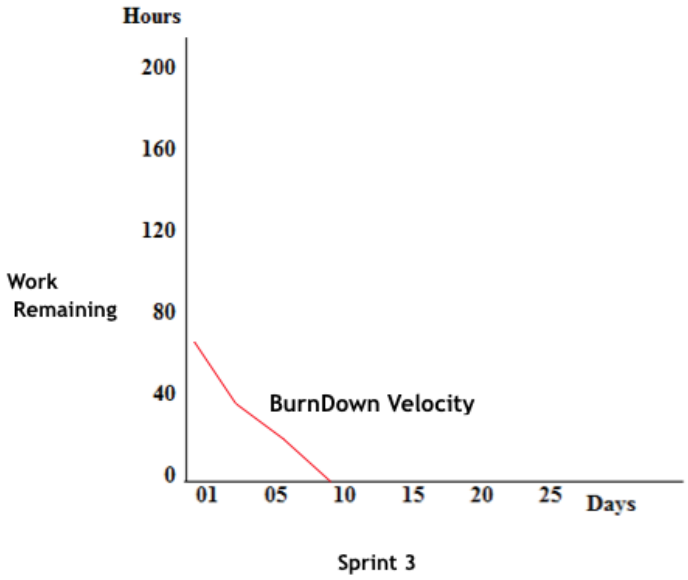


Figure 58. Burn Down Chart for Sprint 3

6.12. Task 10[Test Cases]:

A Test case is a description of a set of interactions between a system and one or more actors (where 'actor' can be people, or other systems). They are usually created as documents, and generally include this kind of information:

- Test case title
- Rationale/description/goal
- Actor/user
- Preconditions (the things that must have already happened in the system)
- Standard path or Main success scenario (what will usually happen, described as a series of steps)
- Post conditions (what the system will have done by the end of the steps).

Our team for solution 404 have completed first sprint backlog and made a test case for those user stories successfully.

We implement test cases for a role of developer, customer, owner as well supplier.

Test Cases for the Customer:

Test Case ID: C1

Test Case Name: Bike Location

Description:

As a Customer, I can get benefit from a system in my app so that, I can locate my bike or e-bike when needed.

Pre-Conditions:

Available Help Desk and Response to Customer

Result:

Test Case #: C1 **Test Case Name: Bike Locating**

System: Information

Designed by: Solution 404 **Design Date:**

Executed by: Agile Team **Execution Date:**

Short Description: Check number of bike And finding location in server

Pre-conditions

GPS device on bike Mobile support GPS Internet connection on mobile.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Lunch application	Show bike location in map	Pass	
2	Enter number of bike	bike number is invalid	Fail	Passed
3	press ok			

Post-conditions

Show bike location in map.

Test Case ID: C3

Test Case Name: Time Selection for Bike

Description:

As a Customer, I would like to get bike for daily and hourly bases so that, I can manage rent of bike according to my usage.

Pre-Conditions:

Available bikes

Result:

Test Case #: C3 **Test Case Name: Select time of rent**

System: Time

Designed by: Solution 404 **Design Date:**

Executed by: Team **Execution Date:**

Short Description: Daily and hourly bases, manage rent of Bike according to usage.

Pre-conditions: Available bike

Step	Action	Actual System Response	Pass/ Fail	Comment
1	Lunch application	Successful process	Pass	
2	select hour or day button	invalid value (more than 3 days or 24 hour)		Passed
3	write value of time		Fail	
4	press OK			

Post-conditions: Accepted by application and show successful pop-up

Test Case ID: C4

Test Case Name: Data Security

Description:

As a Customer, I want that software should keep my personal data confidential and secure so that, nobody else can misuse my data

Pre-Conditions:

Secure Connection.

Result:

Test Case #: C4 | **Test Case Name: Data security**

System: Security

Designed by: Solution 404 | **Design Date:**

Executed by: Team | **Execution Date:**

Short Description: Personal data confidential and secure , nobody else can misuse data

Pre-conditions: secure connection

Step	Action	Actual System Response	Pass/ Fail	Comment
1	Lunch application	Valid PIN and successful payment	Pass	
2	select bike	bike number is invalid		Passed
3	payment step		Fail	
4	receive PIN code before payment			
5	write bank account data			

Post-conditions: Receiving valid PIN from Server and secure payment process

Test Case ID: C5

Test Case Name: Online Chat System

Description:

As a Customer, I need online chat system in software so that, I can communicate directly and immediately and fulfill my requirement.

Pre-Conditions:

Available Help Desk and Response to Customer

Result:

Test Case #: C5

Test Case Name: Online chat system.

System: Information

Designed by: Solution 404

Design Date:

Executed by: Agile Team

Execution Date:

Short Description: online chat system in software to communicate directly and immediately and fulfill my requirement.

Pre-conditions

Available help desk and response to customer

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Lunch application	Customer get the answer and problem will solve	Pass	
2	open chat box	Help desk is not available and customer should wait.	Fail	Passed
3	writing text in text box			

Post-conditions

Available help desk and response to customer.

Test Case ID: C7

Test Case Name: Multi Language Support.

Description:

As a Customer, I want a multi-language supported app for the foreigner

So that, they can operate in their own terminology.

Pre-Conditions:

Check on Multilanguage support

Result:

Test Case #: C7

Test Case Name: Language support

System: Information

Designed by: Solution 404

Design Date:

Executed by: Agile Team

Execution Date:

Short Description: multi-language supported to Operate in their own terminology.

Pre-conditions
Check on multi-language support

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Lunch application	Language changed	Pass	
2	Login successful	Language not changed	Fail	Passed
3	Select the language			

Post-conditions
Change of language when different language is selected.

Test Cases for the Developer:

Test Case ID: D1

Test Case Name: Customers satisfaction (online payment)

Description:

As a Developer I need clear and appropriate user requirement so that, I can develop software accordingly.

Pre-Conditions:

To give easy credit transfer service to the customer.

Result:

Test Case #: D1

Test Case Name: Customers satisfaction (online payment)

System: Information

Designed by: Solution 404

Design Date:

Executed by: Agile Team

Execution Date:

Short Description: Appropriate user Requirement to develop software accordingly.

Pre-conditions

To give easy credit transfer service to the customer.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	log in apps	transfer credit	Pass	
2	Open money transfer portal	Receive confirmation		Passed
3	Put account info		Fail	
4	Transfer money			

Post-conditions

Money Transfer and confirmation

Test Case ID: D2

Test Case Name: Developer Expectation.

Description:

As a Developer, the whole team should be aware of the needs of customer exactly so that no mistake was done while developing a software

Pre-Conditions:

User friendly apps.

Result:

Test Case #: D2

Test Case Name: Developer Expectation.

System: Information

Designed by: Solution 404

Design Date:

Executed by: Agile Team

Execution Date:

Short Description: No mistake was done while developin software, because while developing a software all Features would be inter-related.

Pre-conditions

User friendly apps

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Log in successfully	successfully login	Pass	
2	Test the options are working properly	successfully log off	Fail	Passed
3	logout			

Post-conditions

Smoothly use apps without any problem.

Test Case ID: D4

Test Case Name: Feedback

Description:

As a Development Team, we need to be able to contact the supplier, owner so that we can question them

Pre-Conditions:

On time direct contact or feedback.

Result:

Test Case #: D4 **Test Case Name:** Feedback

System: Information

Designed by: Solution 404 **Design Date:**

Executed by: Agile Team **Execution Date:**

Short Description: able to contact the supplier, owner to question them.

Pre-conditions

On time Direct contact or feedback.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Lunch application	get feedback successful	Pass	
2	contract option	Not get any feedback.	Fail	Passed
3	Massage or chat and phone service			
4	write or call instant			
5	logoff			

Post-conditions

To get feedback instantly.

Test Cases for the Supplier:

Test Case ID: S1

Test Case Name: Developer Expectation.

Description:

As a Supplier, I need the software should be flexible enough to provide user information so that, I can make supply accordingly

Pre-Conditions:

Provide user information to system

Result:

Test Case #: S1

Test Case Name: Check on accurate provision of user information to the system

System: Information

Designed by: Solution 404

Design Date:

Executed by: Agile Team

Execution Date:

Short Description: software should be flexible enough to prov information to supply accordingly.

Pre-conditions

Providing user information to the system

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Lunch application	System receives new data on new user.	Pass	
2	Complete registration	System doesn't receive any data on user.	Fail	Passed

Post-conditions

System receives the complete and accurate information on user

6.13 Task 11[Planning Poker]:

Planning Poker is an agile estimating and planning technique that is consensus based. To start a poker planning session, the product owner or customer reads an agile user story or describes a feature to the estimators.

Each estimator is holding a deck of Planning Poker cards with values like 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40 and 100, which is the sequence we recommend. The values represent the number of story points, ideal days, or other units in which the team estimates.

Our team solutions 404 have estimated each user stories on behalf of the development team. Using this traditional method, we found our estimates were often higher or lower than actual time taken for the task. While these would often balance out, better decisions could have been made if estimates were more accurate.

We have found planning poker to be a rather neat idea and worth getting to know. The core concept brings the project team together to estimate on tasks. This team can include developers, project managers, user experience, designers and sometimes even the end user.

7. Our Estimation:

We had played planning poker for estimating the each user stories by given the points accordingly. And as result, we have found that Customers user stories C1,C4,C5 and C6 has more important to fulfill first and we have given more than 13 points for those 4 user stories. Furthermore, For the rest of the customer user stories C2, C3, C7, C8,

developer user stories D1, D2, D3, D4, Owner user stories O1,O2 and Supplier user stories S1 and S2 , we gave the points between 5 to 8.

Following figure 59 shows the overall estimation for each user stories with story points respectively.

User Story	C1	C2	C3	C4	C5	C6	C7	C8
Estimated Point	20	8	8	20	20	13	8	5
User Story	S1	S2	O1	O2	D1	D2	D3	D4
Estimated point	5	8	8	5	8	5	8	5

Figure 59. Planning Poker Estimation

Appendix: Budget breakdown

NO.	TITLE	COST/PRICE
1	**Scrum Team (Product Owner, Scrum Master, designer, Developer, Tester)**	**30,000 €**
2	**All bikes (1o bikes in each store, so 10*7=70 bikes)**	**28,000.00 €**
3	**accessories**	**7,000 €**
4	**maintenance**	**7,000 €**
5	**System set up**	**25,000**
	Server computers 3 (WEB + customer data storage+ backup)	
	Google Maps	
	Hosting	
	PCI compliance certification	
	Antivirus + Firewall	
	NAGOIS system monitoring	
	work stations for employees	
	Bike embedded GloNav GNS4540 chip system	

Appendix: Responsibility Matrix

Task	Task title	Done by
1	Interview people	All members
2	Define Personas	All members
3	Empathy map for personas	All members
4	Define user stories	All members
5	Create non-functional prototype	Mobile application: Murad Eminov – Software: Chandni Jariwala, Amir Doorandish
6	Interview people about prototype	All members
7	Changed user stories	Murad Eminov
8	Building product backlog	Chandni Jariwala
9	Building sprint backlog	Chetan Kumar Doberive
10	Test case for an acceptance test	Murad Eminov, Amir Doorandish, Khaled Mosherraf
11	Planning poker	All members

REFERENCES

[1]. Wesley , Addison, 2012. Essential Scrum. *Essential Scrum*, 3, 15.

[2] [Personas: The Foundation of a Great User Experience | UX Magazine. 2015. Personas: The Foundation of a Great User Experience | UX Magazine. [ONLINE] Available at: http://uxmag.com/articles/personas-the-foundation-of-a-great-user-experience. [Accessed 24 June 2015].]

[3] Empathy Map: step into your client shoes. 2015. Empathy Map: step into your client shoes. [ONLINE] Available at: http://blog.leanmonitor.com/empathy-map-step-client-shoes/. [Accessed 23 June 2015].

[4] Scrum Backlog. 2015. Scrum Backlog. [ONLINE] Available at: http://scrummethodology.com/the-scrum-backlog/. [Accessed 20 June 2015].

[5] Sprint Backlog and the Scrum Sprint. 2015. Sprint Backlog and the Scrum Sprint. [ONLINE] Available at: https://www.mountaingoatsoftware.com/agile/scrum/sprint-backlog. [Accessed 19 June 2015].
[6] Measuring Alignment with the Principles of Agile - Scrum Alliance . 2015. Measuring Alignment with the Principles of Agile - Scrum Alliance . [ONLINE] Available at: https://www.scrumalliance.org/community/articles/2014/january/measuring-alignment-with-the-principles-of-agile.aspx. [Accessed 20 June 2015].
[7] What is Agile model – advantages, disadvantages and when to use it?. 2015. What is Agile model – advantages, disadvantages and when to use it?. [ONLINE] Available at: http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/. [Accessed 23 June 2015].
[8] Agile & Waterfall Methodologies - A Side-By-Side Comparison. 2015. Agile & Waterfall Methodologies - A Side-By-Side Comparison. [ONLINE] Available at: http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/. [Accessed 15 June 2015].
[9]Frameworks & Standards Reality Check: Let’s ask Google Trends | | Vom Kennen zum Können zum Tun. 2015. Frameworks & Standards Reality Check: Let’s ask Google Trends | | Vom Kennen zum Können zum Tun. [ONLINE] Available at: http://blog.itil.org/2014/09/allgemein/frameworks-standards-reality-check-lets-ask-google-trends/. [Accessed 12 June 2015].
[10] Abdallah Qusef, Andrea De Lucia, 2010. Requirements Engineering in Agile Software Development. JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, [Online]. VOL. 2, NO. 3, 212. Available at: http://www.unisa.it/dipartimenti/dip_informatica/index [Accessed 11 June 2015].