Decision Tree Engine App

Project Plan

2014 - 11 - 24

Group 3:

Valerio Lucantonio, Julieth Castellanos,Nathan Chape,Tamara Dancheva,Anna Enbom,Eric Johansson, Niklas Sjöqvist

1 Introduction

Decision Tree App Engine is the project assigned to our group, group 3, for Software Engineering Project Teamwork course at Mälardalens Högskola. The purpose of this project is to create an application that helps users to make investment decisions - in different fields (economics, automation, etc..). To this end, the application will utilize *decision trees*. End users will get statistics from the survey on how many of the questions answered ”approve” an investment or how many that ”don’t approve” and how many that still are ”uncertain”. The application must also offer the application admin the possibility to create these surveys/trees. This document will go through a presentation of the group, our organization and used tools, planned effort and deliverables, quality assurance (section 2), followed by the technical section. Background, architecture, functional and nonfunctional requirements, traceability validation and verification (section 3).

2 Project Organization

Project manager

Valerio Lucantonio has been assigned this position and will therefore take a leading role within the team to help coordinate tasks and to help initiate decision making. He will also function as the groups main spokesperson towards the steering group.

Customer relations

Anna Enbom volunteered for this position and has since been responsible for communication with the client and for setting up meetings with the client.

Github maintenance

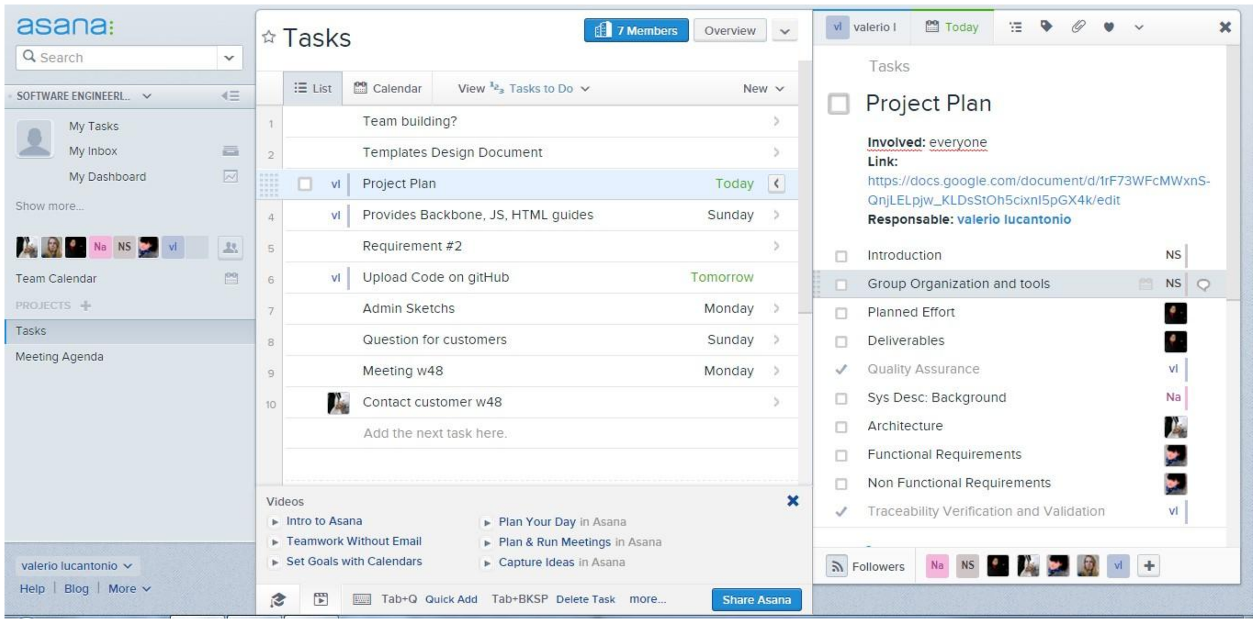
Tamara Dancheva is responsible for setting up and maintaining the group’s GitHub repository.

Development team

All project members are part of the development team. The purpose of the team is to design, implement and test the application. They are all responsible for assigning themselves tasks, ask for help from other members when they’re in need and to actively participate in the project. All previously listed members and Julieth Castellanos, Nathan Chape, Eric Johansson and Niklas Sjöqvist will be part of the development team.

2.1 Group Organization and tools

Communication within the group is done through email and the facebook message system. The group shares documents between each other on Google-drive and three different backlogs will be stored on ASANA[5]. One for tasks that are to be implemented in the application which are done on an individual level. Another for team coordination containing tasks that affects the entire team and a final one which is used to create and plan meetings and meeting agendas. These backlogs are divided into separate folders. Code is shared through Github's online services and documents that are to be shared with the client are done so by using dropbox.



1		Team building?	
2		Templates Design Document	
3	vl	Project Plan	Today
4	vl	Provides Backbone, JS, HTML guides	Sunday
5		Requirement #2	
6	vl	Upload Code on gitHub	Tomorrow
7		Admin Sketchs	Monday
8		Question for customers	Sunday
9		Meeting w48	Monday
10		Contact customer w48	

Add the next task here.

Tasks

Project Plan

Involved: everyone
Link: https://docs.google.com/document/d/1rF73WFcMWxnS-QnjLELpjw_KLDsStOh5cixnl5pGX4k/edit
Responsable: valerio lucantonio

☐	Introduction	NS
☐	Group Organization and tools	NS
☐	Planned Effort	
☐	Deliverables	
✓	Quality Assurance	vl
☐	Sys Desc: Background	Na
☐	Architecture	
☐	Functional Requirements	
☐	Non Functional Requirements	
✓	Traceability Verification and Validation	vl

Fig.1: Asana backlog: Organization of this document

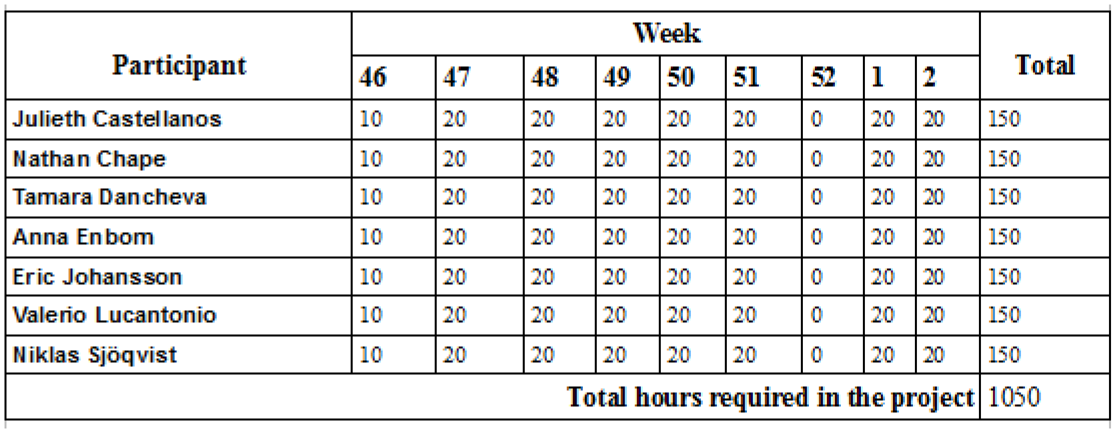
Meetings are organized and planned through the use of ASANA[5] which, in turn, is linked to our respective e-mail. When needed, meetings can also be planned through the facebook message system where the group has created a chat room for quick communication.

Work hours and results are to be presented verbally or in text at a weekly meeting when the presentations for the steering group are constructed, these meetings are held a couple of hours in advance of meeting with the steering group. If a member is unable to do so they should ahead of time notify a team member (preferably the project manager) about this and give them the information required for them to present their work hours and results.

Backbone[3] and Cordova[4] are both planned to be used to develop the application. The team will be using pair programming[6] as often as possible to simplify the process of divisioning the labour. During the implementation process it’s highly likely that a few members will take respectively leading roles in different fields reflecting their areas of knowledge such as user testing, prototyping and continual design.

2.2 Planned effort

The following table represents the estimated weekly planned effort for each member of the group.



Participant	Week									Total
	46	47	48	49	50	51	52	1	2	
Julieth Castellanos	10	20	20	20	20	20	0	20	20	150
Nathan Chape	10	20	20	20	20	20	0	20	20	150
Tamara Dancheva	10	20	20	20	20	20	0	20	20	150
Anna Enbom	10	20	20	20	20	20	0	20	20	150
Eric Johansson	10	20	20	20	20	20	0	20	20	150
Valerio Lucantonio	10	20	20	20	20	20	0	20	20	150
Niklas Sjöqvist	10	20	20	20	20	20	0	20	20	150
Total hours required in the project										1050

Fig.2: Planned Effort

2.3 Deliverables

Course deliverables	**Date**
Project plan	20/11/2014
Design description (first version)	4/12/2014
First prototype	4/12/2014
Test specification	18/12/2014
Design description (final version)	8/12/2014
Final prototype	8/12/2014

Client deliverables	**Date**
Requirements document (first version)	20/11/2014
First prototype	4/12/2014
Final prototype	8/12/2014
User Manual	12/12/2014

2.4 Quality Assurance

In this analysis quality assurance should be guaranteed from three points of view:

- Product quality: we will ensure the (internal) quality of the product in regards to the engine and the algorithms. For the front-end, we will have several meeting with the customers, in which we will demonstrate mock ups and sketches of the application design in order to get feedback from the client.
- Deliverables quality : we will ensure the quality of deliverables using our previous knowledge acquired during other courses of software engineering. Including documentation and templates from previous course.
- Development process quality: we are trying to achieve a good quality of the development process by having at least two meetings per week, and continually trying to reach a consensus to the most pressing problems. Moreover, we are using several tools (Asana, Google Doc, Dropbox) to share all the possible information to keep the group members' aware of the current status of the project. In this course the time has a very crucial roles, and usually time and quality are divergent. In order to respect deadlines and to deliver all the required materials we are organizing documents with Asana assigning the task to one among us that will be the responsabile and dividing the sections of the document and assigning them among us. For the implementation we will make three sub-groups of two people implementing the same feature and all the features will be managed in a product backlog in asasa with the same pattern of the document.

3 System Description

3.1 Background

As stated in the introduction, our goal is twofold:

- to develop a web based application that allows an administrator to easily create *decision trees* for various applications, e.g. Economic analysis, Industry and Production, Automation… etc.
- and secondly, to develop a mobile application that allows a end users to access these trees, and answer some questions about investments in their respective domains. The application should be available for Android and IOS devices.

The application for the administrator will be deployed for one administrator, and it will be a web application (probably running locally). Instead the application for end users will be android and ios applications.

3.1.1 Existing System

The client has stated that they have, internally, developed an excel based application that they use to develop decision trees. Whilst conceptually, a basis for the current application we are developing, it is unlikely there will be structural - beyond the formal decision tree structure - similarities between the two.

In regards to the end user interface, the client has supplied us with a *user interface* demonstration that we could possibly use in our development of the mobile application. While data storage, in some form (e.g. database), will be necessary for testing. It is currently unclear how the client intends to store data used and created in the final product.

3.2 System Overview

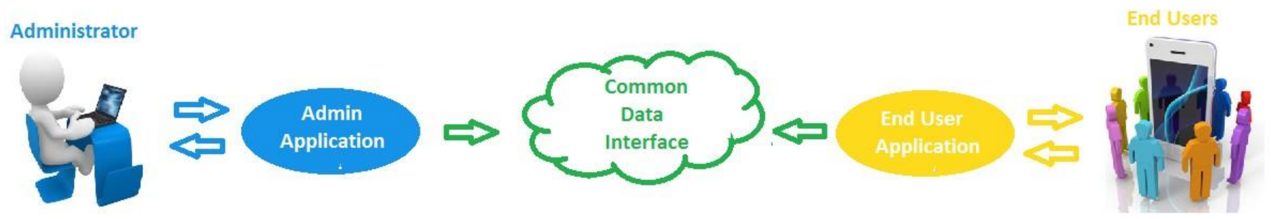


Fig.3: Overview

We will develop a decision tree engine application for the administrator to develop trees. According to the customer, there will only be one administrator (Andreas) that will develop all the trees and he will be using the admin application through a desktop computer or laptop. There will also be an application for end-users. The end-user will go through the questions in the tree and receive statistics on how many of the questions answered that ”approve” an investment, how many that ”don’t approve” and how many that still is ”uncertain”. This is in percentage %. Easiest way is by putting weight on the answers.

We have decided to develop the end-user application in Cordova. Cordova is a hybrid cross-platform tool for mobile applications. It is called hybrid because it can be deployed like a native app (for Android and IOS devices) but it is made like a web page (HTML, CSS and JavaScript) and executed in a browser environment. The advantages of an approach like this are that the same code can be used for all mobile platforms and almost all the native functionalities can be accessed. Maintenance is also assured because the source code is only one and every operations of updating will produce consistent final apps. We have also decided to use the Backbone.js as a business logic layer. Backbone.js is a lightweight framework. Some call it a MVC (Model View Controller) framework, some call it a MV* framework.

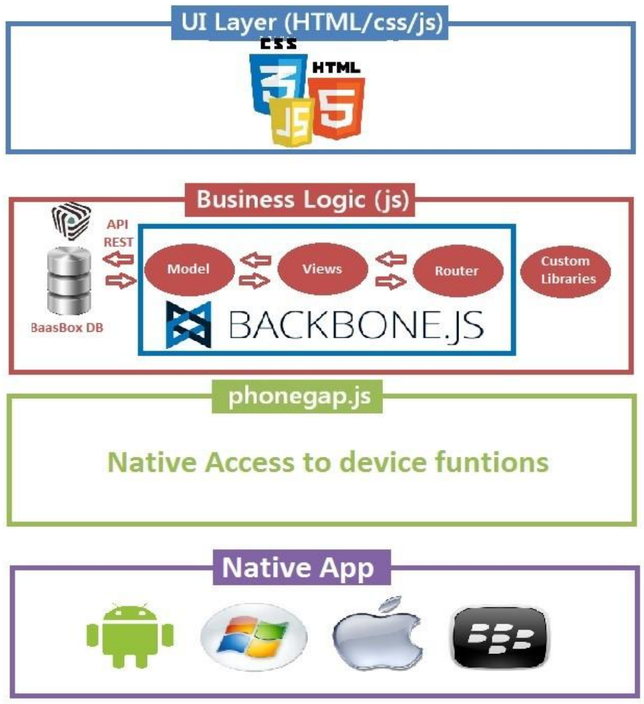


Fig.4: Cordova Architecture

The admin tool will probably be a web application as well, so that we can reuse as much code as possible. We will use Backbone outside Cordova to be consistent with the front end application and to make reuse of the tree structures. We are still not sure how the applications will communicate each other because we need yet to discuss this with the customer, but we have two different proposal: local or remote DB.
We have not yet decided on the structure of the tree and how to store trees. We made a simple sketch on how it might look (not a finished solution):

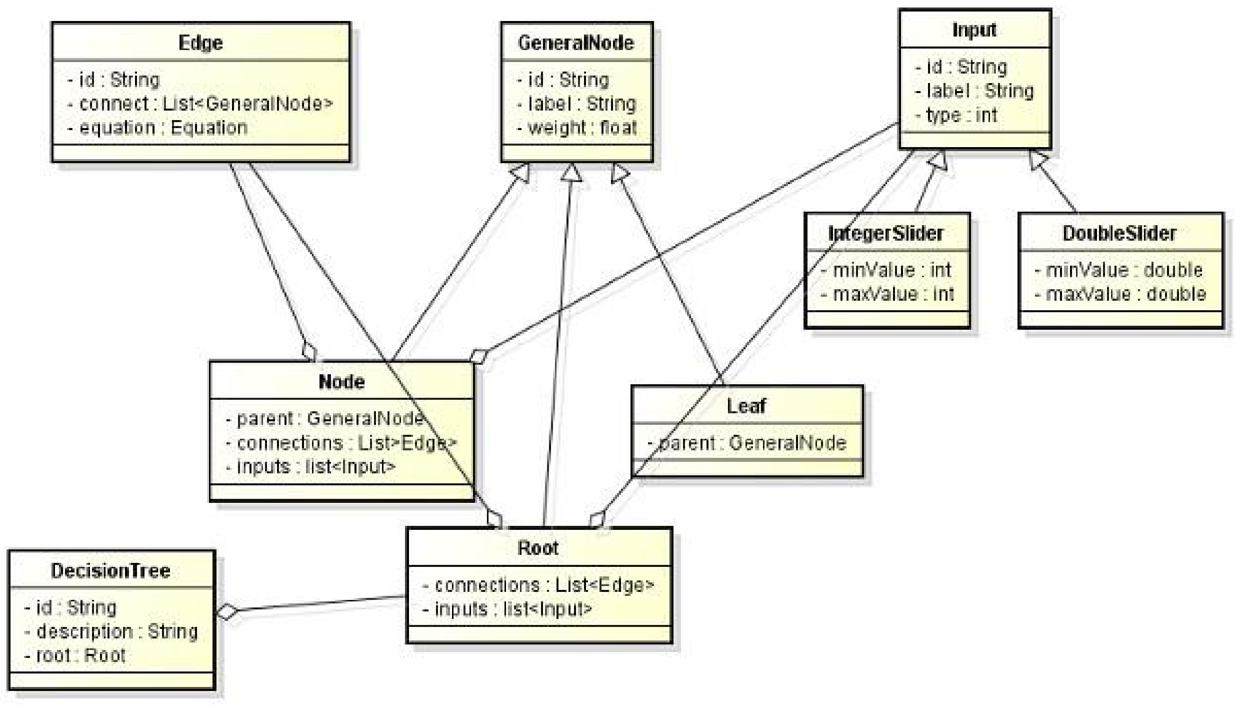


Fig.5: Tree Engine Model

3.3 Functional Requirements

A tree creator/editor, engine, and a web application (into which the engine is integrated) to test the functionality should be developed. There are two main categories of actors in this project: application end users and administrator.

The end users should be able to access the surveys offline. They should be able to continue working on a survey they have started or change the answers on a survey they have completed. Once a survey is completed, users should be able to forward it to an email address. Other features include changing the language, subscribing for updates, having access to tips and advices and a general description of the application.

The administrator uses the tree creator/editor to create and edit trees that are interpreted by the end-user application, add and edit questions and answers and add new languages. The administrator should be able to add weight to each node, as well as a statement (advice/tip) associated to each answer for each question and to add groups of answers that are used to evaluate the survey(positive, negative, indeterminate).

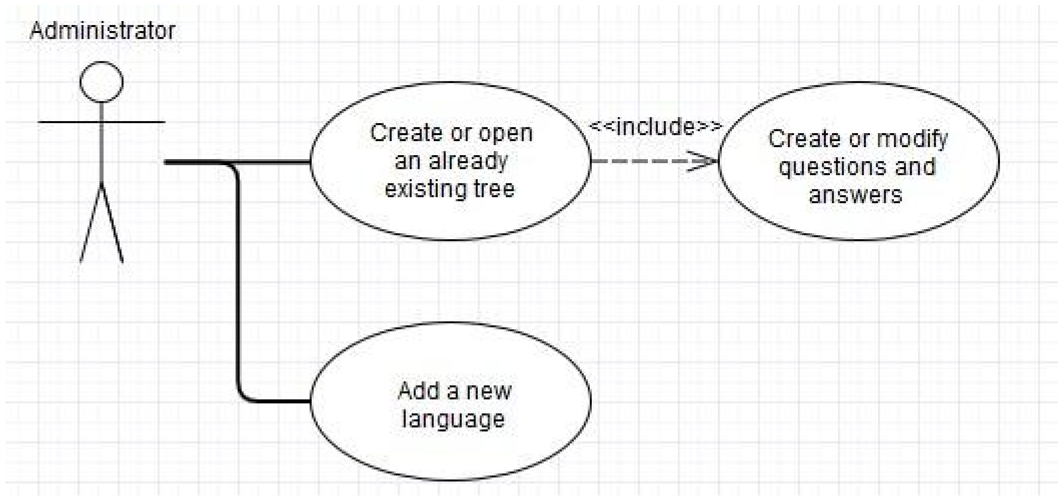

Administrator
Create or open an already existing tree
<<include>>
Create or modify questions and answers
Add a new language

Fig.6: Admin Use Case Diagram

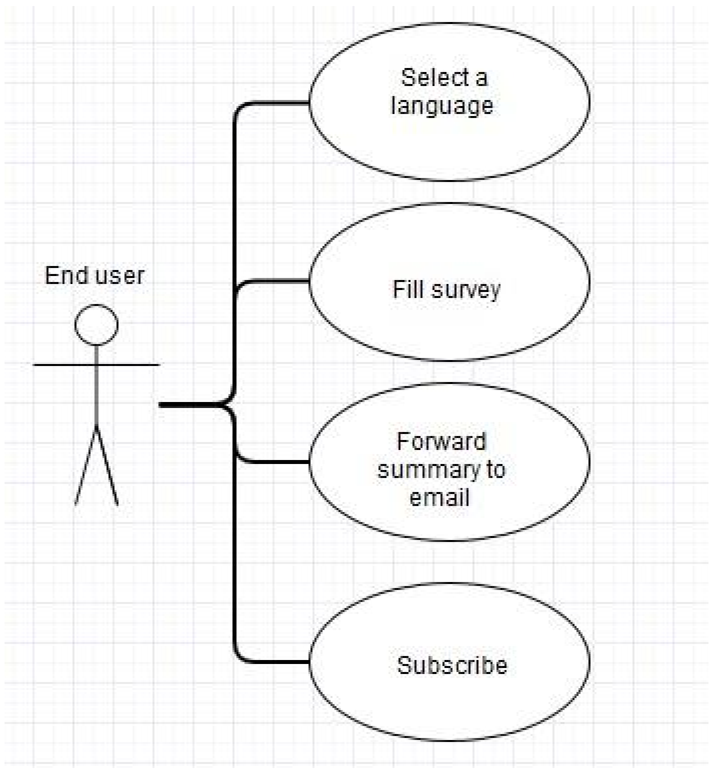


Fig.7: End User Use Case Diagram

3.4 Product Backlog

The following table represents the backlog. Each activity is defined by a *number* (ID), the *priority* (value from 1 to 5). The estimated *effort* (in terms of hours) and the *type* of the activity (if the activity belongs to a particular phase: planning, documentation, design, client documentation, applications implementation).

#	Activities	Priority	Type	Effort
1	Group Organization and Communication Plan	5	Planning	10
2	Define meeting with customer	5	Planning	10
3	Define technologies	5	Planning	10
4	Project Plan Document	5	Documentation	30
5	Requirements Collecting	5	Requirements	50

6	Design Static Architecture	5	Design	60
7	Design Dynamic Structure	5	Design	50
8	Prepare Mokeups	3	Design	25
9	Design Description Document	5	Documentation	70
10	Test Specification	5	Documentation	40
11	Project Report	5	Documentation	50
12	User Manual	3	Client Documentation	25
13	Code Documentation	3	Client Documentation	25
14	Define Tree Models	5	Both Apps	45
15	Define Database Tree Structure	5	Both Apps	25
16	Define Rest API	5	Both Apps	10
17	Add, Modify, Delete Survey	5	Admin App	70
18	Add Survey Language	4	Admin App	50
19	Add, Modify, Delete Questions	5	Admin App	90
20	Add, Modify, Delete Answers	5	Admin App	90
21	Multiple Language Support	3	End-User App	50
22	Choose Survey	5	End-User App	35
23	Answer Survey	5	End-User App	70
24	Generate Summary	5	End-User App	70
25	Save Survey State	4	End-User App	60
26	Forward Summary	4	End-User App	55

3.5 Non-Functional Requirements

3.5.1 Usability

The applications targeted for both the users and the administrators should be simple, highly intuitive and easy to understand. The graphical user interface should be non-distracting, with easily readable font. From an aesthetic point of view, the look of the application does not have to be embellished.

The administrator application should be implemented in a way so the admin with no programming skills can easily create and edit decision trees. A clear, easy to edit view of the tree has to be provided. Drag and drop is desirable.

As for the user’s application, the application should contain information about how to use the application and an option to check out a question’s specific description, making it more user-friendly.

3.5.2 Reliability

The reliability of the decision tree engine is of great importance. The engine should always produce a correct tree structure and correct end results after completing the survey according to the formula provided by the administrator who created the survey.

3.5.3 Testability

Testability is necessary in order to make sure that the engine is reliable and always producing the correct output. Therefore modularity of the code and separation of concerns are essential, the modules have to be as isolated as possible to ease the process.

3.5.4 Portability

Portability is one of the main requirements in this projects since the goal is to make the user application accessible for as wider audience as possible deploying it in a multiple different platforms (Android, IOS).

3.5.5 Reusability

The engine source code should be independent of the user interface, therefore making the code easily reusable. This is one of the main requirements for this project.

3.5.7 Extensibility

Since work on the engine will probably be resumed after the hand-in of the code, it is very important that the code leaves space for an easy way of adding new features and modification of existing features.

3.5.8 Documentation

In order to assure reusability, maintainability and extensibility, extensive documentation has to be provided including: project plan, design description, test specification, project report. The source code has to be easily comprehensible, including a lot of comments , sufficient code documentation and has to be written following a set of code standard rules.

3.5.9 Price

The project’s prototype has initially been developed using the Xamarin Forms framework. Because of the high price for the license, a cheaper alternative for creating a cross-platform solution is sought. Using open-source frameworks like Cordova and libraries like Backbone is therefore a suitable choice.

3.5.10 Performance

The performance is not of great importance in this project.

3.6 Verification and Validation

We will manage these two important aspects of this project in following way:

- Verification: in order to verify that the application meets the requirements and that the process of building it is correct, we will follow the strict pattern provided from the technology/tools. We will also divide our work and follow backbone’s patterns systematically to assure the product will be built correctly.
- Validation: we will validate our product by having weekly meetings with customers to see if the product meets his expectations. The validation process will start as soon as the requirements definition is completed and used continually throughout the development process. As we are familiar with the agile approach we know that the process of validation is important, and that work of an entire week could be meaningful. To manage this problem we need to be sure, before we start coding, that the expected result and final result will be the same. We will manage this with granularity on features in that we will start to implement a new feature *only* if both the logic and the presentation layers are in accordance with the clients wishes.

References:

[1] Pressman, R. S. (2010). Product Metrics. In *Software engineering: A practitioner's approach* (7th ed., pp. 620–623). New York: Mc Graw Hill.

[2] "What is Eclipse?". Eclipse.org. Retrieved 2014-11-20.
http://help.eclipse.org/luna/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Fint_eclipse.htm

[3] "BACKBONE.JS" 2014-02-20. Retrieved 2014-11-20. http://backbonejs.org/

[4] "About Apache Cordova^{TM}". Retrieved 2014-11-20. http://cordova.apache.org/

[5] "LEARN The ASANA Basics". Retrieved 2014-11-20. https://asana.com/guide/learn

[6] "Pair Programming"Retrieved 2014-11-20.
http://www.extremeprogramming.org/rules/pair.html