MÄLARDALENS HÖGSKOLA
ESKILSTUNA VÄSTERÅS

PROJECT PLAN REPORT

Project name: Solar energy calculator
Group number: 2
Date: 2016-11-17
Course: Software Engineering 2: Project Teamwork (DVA313)
Academy: Innovation, design and technology

1. Introduction

We are going to do a project for the client named Bengt Stridh from the Future Energy Center research specialization at Mälardalen University. The client and his colleagues has in a previous project developed detailed models that are used to analyze investment decisions for photovoltaic (PV) plants in Sweden. Our task in this project is to develop a web-based tool to support different stakeholders, such as private persons and companies. The users should in a straightforward fashion be able to determine what investments in solar energy that are suitable for them, based on a number of default parameters that can be adjusted by the user. The photovoltaic market has been growing strongly during the last years. However, the general knowledge of PV among potential investors is still low and therefore they could miss out on for example financial and environmental benefits. Because of this there is a strong need for a user friendly tool to calculate both production cost and profitability for PV investments in Sweden.

2. Project Organization

2.1 Project group (members, contact info, roles or responsibilities)

Name	Email	Roles/Responsibilities
Lukas Hamacek	lhamacek@outlook.de	Project manager
Aliya Hussain	syeda.aliya.hussain@gmail.com	Documentation and presentation
Charlie Höglund	c.hoglund@live.se	Trello manager
Jonathan Larsson	jonathan-larsson@outlook.com	Configuration manager (SVN/Github)
Sebastian Lindgren	sebbestune@icloud.com	Client contact
Avalika Podduturu Reddy	avalika12@gmail.com	Report overseer and designer

OBS. The roles or responsibilities might change during the project (e.g. if a person is more suitable for a role/responsibility). All of the project members will also help out the other project members with their roles/responsibilities if necessary.

2.2 Organization and communication

Every monday during this course we will have a project meeting with the external steering group in the school's facilities. Above that we will have at least one internal project meeting per week, either physically in the school's facilities or non-physically over for example the communication service called slack. We will also make sure to have a small internal project meeting before the meeting with the external steering group on mondays, which will allow us to go through all of the material to be presented to the steering group. The number of internal project meetings may differ from week to week depending on whether we actually need more meetings or not. As mentioned before the forms of contact will either be physical (i.e. in the school's facilities) or non-physical (i.e. over communication services like slack or email). Each project member will report their own worked hours per week by entering the necessary data in an Excel file (i.e. a specific file for each project member) available on the Github repository. All of the results such as documentation and code will be reported by committing/uploading them to the Github repository, which allows both the project members and the steering group to access them. All of the project members will be able to commit results to the Github repository, but sometimes it might be required to have internal discussions with some particular project members before merging for example your code with theirs.

2.3 Planned effort per member for each week in the project

This project course is on 7.5 hp and 50% pace, which corresponds to approximately 20 work hours per week for each project member. Then there is approximately a total of 8 work weeks, since there will be a few days of absence among the project members during the Christmas holidays, the last week of the course will not be a full week etcetera. Therefore, each project member should have a total effort of approximately 160 hours in the end (i.e. 20 hours * 8 weeks = 160 hours). However, the project members may choose for themselves how to divide their total of 160 hours throughout the weeks as long as the internal and external deadlines are fulfilled.

2.4 Deliverables, deadlines, milestones and activities

EXTERNAL

Deliverables:

Deliverable	**Deadline**	**Total effort in hours**
Project plan	Nov 17	35
Design description (1st version)	Dec 1	30
Product (1st version)	Dec 1	100
Design description (final version)	Jan 12	60
Product (final version)	Jan 12	200
Project report	Jan 12	200

Meetings with the steering group (25 min effort each, total effort: ~3 hours/member):
Project meeting - Nov 9
Project meeting - Nov 14
Project meeting - Nov 21
Project meeting - Nov 28
Project meeting - Dec 5
Project meeting - Dec 12
Project meeting - Dec 19

Presentations (25 min effort each, total effort: 1,25 hours/member):
Project plan and requirements - Nov 23
Preliminary design and implementation - Dec 7
Final presentation - Jan 11

INTERNAL

General activities:

Activity	**Deadline**	**Total effort in hours**
Learn how to use Github/Git (i.e. the project members who need it)	Nov 16	10
Learn how to use Trello	Nov 16	15
Build and present a lightweight website layout for the client	Nov 22	30
Learn/relearn the necessary web skills (e.g. php, ajax, javascript etc.)	Nov 28	120

Revisions of the deliverables:

Activity	**Deadline**	**Total effort in hours**
Revision of the project plan	Nov 17	6
Revision of the design description (1st version)	Dec 1	6
User test of the 1st product version	Dec 1	6
Revision of the product (1st version)	Dec 1	6
Revision of the design description (final version)	Jan 12	10
User test of the final product version	Jan 12	10
Revision of the product (final version)	Jan 12	15
Revision of the project report	Jan 12	15

Preparations before the presentations:

Activity	**Deadline**	**Total effort in hours**
Prepare for the “project plan and requirements” presentation	Nov 23	15
Prepare for the “preliminary design and implementation” presentation	Dec 7	15
Prepare for the “final presentation”	Jan 11	20

2.5 Quality assurance

Firstly, we have the quality assurance of the documentation deliverables such as the project plan, design description, project report and so on. Before reporting the specific documentation deliverable, we plan to have an activity where at least one project member read through the documentation again while looking for grammatical, layout or logical mistakes etcetera. Secondly we have the quality assurance of the implementation deliverables such as the finished code and graphical user interface (i.e. the final product). To make sure that the graphical user interface is for example user-friendly, then we can let both the client and other external people (e.g. students on the school) do a user test, where they can try it out and share valuable feedback. To assure the quality of the different units (i.e. functionalities/features) of the system we will start with doing unit testing, which will be done as soon as each unit is implemented. Then we will do an integration test to assure that all of the different units of the system works together as expected. Finally, we will test the complete system by letting both the client and other external people try it out, which will allow us to see if the system works as expected in a real life scenario.

3. Description of the system to be developed

3.1 High level description of the domain and the problem

A preliminary specification has been provided by the client. The purpose of the calculator is simply to calculate the production cost and profitability for photovoltaic installations in Sweden. Its intended users are both private persons and other users (e.g. companies, property owners). The problem is that very few users have enough knowledge about PV and that there are a lot of necessary parameters to be inputted for the calculations. However, the solution for this problem is by using default parameters (i.e. different for private and other persons), min and max values, guiding texts etcetera. The web-tool should also preferably have the following functionality: Save used input values from one session to another, switch between Swedish and English language, compare the calculated values for two sets of input parameters. The web-tool's output should be production cost, profitability, cash flow, diagrams (i.e. with present values, cost shares and income shares). There should also be a possibility to make a report file that could be printed with input and output values. An important feature of the web calculator is to keep it up to date with legalizations, prices of electricity, prices of electricity certificates etcetera. Therefore, it should be easy to update guiding texts, default values, as well as suggested minimum and maximum values. If possible the updates of the system should be made by uploading an Excel file containing the updated values.

3.2 Description of existing systems

The system that is currently being used is in the form of an Excel file, in which the user has to enter the necessary parameters inside cells and then some built-in functions calculate the result. Our own web application will at least use the same functionalities as the ones in this already existing system, but bring it to the web. This new system will benefit from the dynamical nature of web applications, hopefully enable a more user friendly environment, and also make the photovoltaic investment calculator accessible to a larger audience. In other words, the already existing system will not be extended, but instead give us knowledge about the functional basis on which our web application will be built upon.

3.3 High-level description of the desired functionality

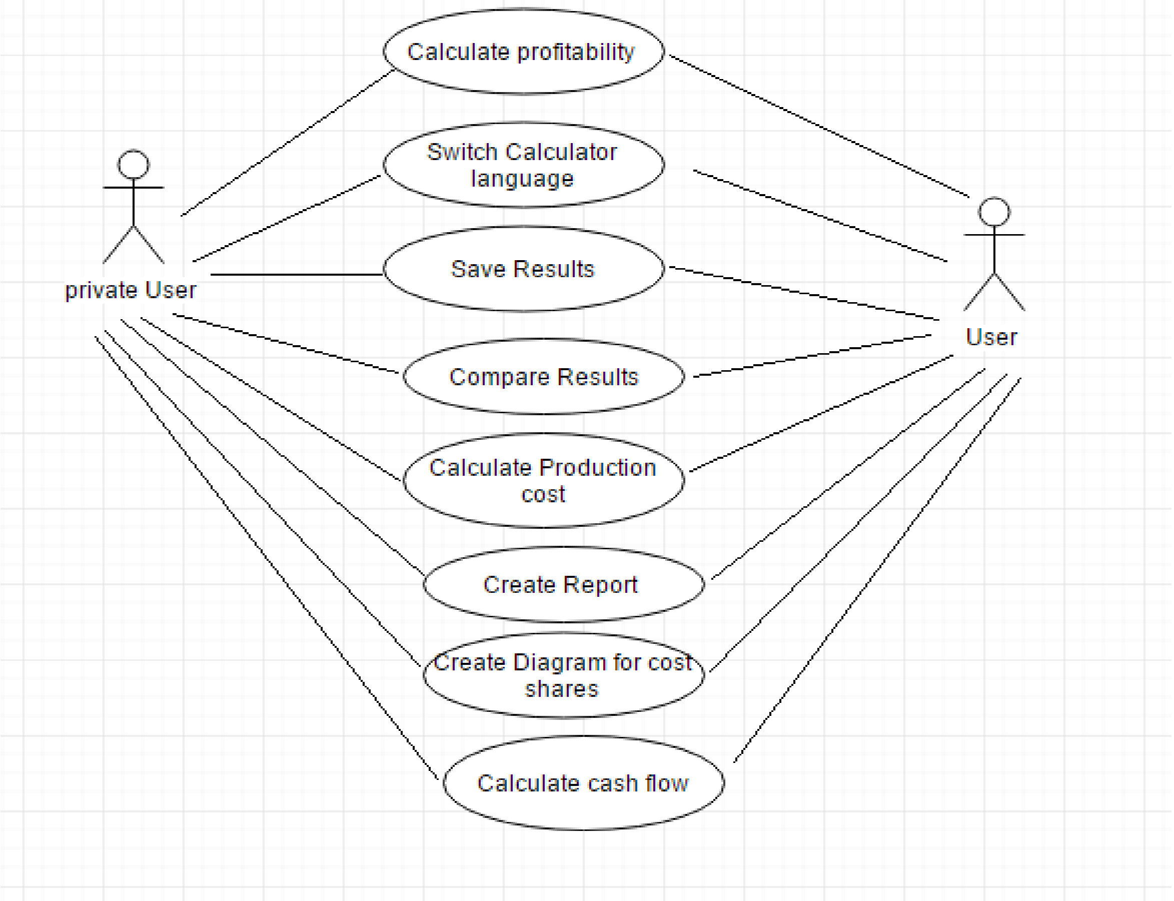

Figure 1- This is a use-case diagram that captures the desired functionality. The actor “private user” is for the private persons, and the actor “user” is for the other persons (e.g. companies).

4. Initial project backlog

Feature	**Estimated time (Planning Poker 0, 1, 2, 3, 5, 8, 13, 20, 40, 80, 100)**
Bootstrap Design/Layout	80
Database structure (ERM-Model)	13
Database implementation	20
Create a new account	40
Create a admin account to update values	20
Update function for admin	20
Understand the formula and implement it	80
Create automatically a PDF file with input/output values	40
Send the output file per email	8
Help text for parameters	2
Switch between English and Swedish	13
Compare to calculated values	13
Output (Cash Flow, Diagrams, …)	40