Project Plan for Bicycle Garage Pro (Group 33, 2015)

Current version: 1.0.0

Alexander Skafte
tfy13ask@student.lu.se
Dennis Jin
desuvader@gmail.com
Petter Berntsson
dat14pbe@student.lu.se

Contents

1 References

- *Examples and Exercises in the Software Engineering Process*. ETSA01 VT 2015. Deparment of Computer Science, Lund University. March 10, 2015.
- *Software Requirements Specification for Bicycle Garage Pro*. ETSA01, Group 33, 2015.

2 Introduction

2.1 Project model

2.2 Purpose

The aim of the project is to develop software for a bicycle garage which provides safe storage of bicycles. The software will be developed for prespecified hardware.

2.3 Goals

2.3.1 Product goals

The software shall be able to securely handle users checking in and out their bicycles of the garage.

The product will take care of users passing through the entrance with their bicycle - the user will enter a PIN code and scan a barcode for their bicycle to store it. The user can then exit the garage, and to exit with a bicycle the same procedure will be done as the one to store it.

For a more detailed overview of the product, please refer to *Software Requirements Specification for Bicycle Garage Pro*.

2.3.2 Business goals

To offer cheap and reliable bicycle storage to customers, whom may want to protect their bicycle against misfortune and use the bicycle instead of public transport of their cars.

2.4 Limitations

Hardware limitations

Since the software must make use of prespecified hardware, there is a limit set by factors other than that of the software's implementation that will decide how effective the system will be at doing its job. If there are fundamental flaws in the hardware, the software too will suffer. To counter this potential issue, thorough testing must be implemented in the software.

Temporal limitations

In addition, there is a clear deadline for the project. This may result in an unfinished and/or underdeveloped product being released, which may result in expensive bugs and a bad reputation should the software fail its customers.

Economical limitations
The budget is limited, and thus more developers or other personnel cannot be hired.

3 Project Organization

3.1 Development organization

Responsible for for this project are:

Alexander Skafte:

- Requirements specification
- Test plan
- Design document
- Project plan
- Various reviews
- Software development

Dennis Jin:

- Requirements specification
- Test plan
- Design document
- Various reviews
- Software development

Petter Berntsson:

- Project plan

3.2 Stakeholders

- **The municipality** – For environmental reasons; if an increased number of citizens use their bicycles instead of their cars or public transport, the greenhouse effect can be lessened.
- **ACME** – The company which manufactures the bicycle garages.

4 Hardware and software resources

LaTeX: Typesetting software; used to typeset all documents related to the development of *Bicycle Garage Pro*.

Eclipse: Integrated desktop environment for software development.

The Java Programming Language: The programming language used to create the software.

Google Drive: Cloud storage reachable for all members to edit.

Git & GitHub: For sharing code and LaTeX source text.

5 Division of labor

5.1 Activities

Project plan
The planning of the project; also this rapport.

Requirements specifications
The requirements for the project are identified and defined. This results in a requirements specification.

Test plan
The planning of all tests that will be performed. This results in a test plan.

Design
A general description of the systems structure. This results in a design document. Different people are responsible for different parts of the code.

Implementation and unit testing
Implementation and testing of all parts of the system. This is performed according to the design.

Integration
This part is the finishing part of the program, which leads to a fully functioning program.

System test
The system is tested in its entirety. This takes part in a testing environment which mimics the environment that the program will be used in.

5.2 Deliverables

- Project plan
- Requirements specification
- Design document
- Test plan
- Source code
- Runnable JAR file

5.3 Schedule and estimated work

Below is a schedule showing the time estimated for the project's different parts. The colors carry no significance.

Week	M	T	O	T	F	L	S	Comments
13								
14								
15								More work required from each individual team member.
16	Delivarable 3			Seminar	Group disc.	Finish spec. Start tests.		Review feedback. Finish the spec. in good time and start working on tests.
17				Seminar	Group disc.	Finish tests. Start prog.		Finish tests and spec. and start programming. Build design document as you go. Clean up spec. before we have to start filing things for change.
18	Delivarable 4				Group disc.	Finish prog. Start clean-up		Finish up the program and design documents.
19	Delivarable 5.a) Test plan, Design				Group disc.	Clean-up		Review everything we've done so far and clean-up the slack.
20	Delivarable 5.b)				Group disc.			All this extra time should be dedicated to cleaning up.
21								Wrap-up week. Finish everything before Delivarable 6.
22	Delivarable 6							
23								
Delivarable 3	Req. spec. 0.99, Project plan							
Deliverable 4	Req. spec. 1.0, review protocol							
Deliverable 5a	Test 1.0, Design 1.0							
Deliverable 5b	QA							
Deliverable 6	Everything complete							

6 Report, follow-up and quality assurance

To make collaboration easy when writing both the software and the project related documents, the version control system *Git* will be used, with the files hosted on the free Git hosting platform *GitHub*. In addition, both internal and external reviews will be conducted in a common *Google Drive* folder, using the *Google Docs* service.

All PDF versions of the documents will be found in the Google Drive folder when they need to be there. The latest versions shall always be available on GitHub.

During the seminar sessions provided, the group will gather and discuss further plans for the project, as well as meet with the tutor who will provide feedback on the documents handed in through the Google Drive folder.

Communication within the group will be conducted through a social media platform previously agreed on, or through email if necessary. The reason for primarily using a social media platform for this purpose is that communication is faster and more interactive than if one were to solely use emails. Additionally, commit messages on Git shall be clear and provide a third means of communication.

A Risk analysis

Risk case 1:	**The hardware does not comply with its specification.**
Risk:	Low.
Effect:	Devastating.
Measure:	Contact the hardware developers and ask for new hardware specifications.
Risk indicator:	The appliance does not work properly, even if the virtual testing passed.

Risk case 2:	**Unwanted document modification or deletion**
Risk:	Possible
Effect:	Potentially devastating
Measure:	Use a version control system (such as Git).
Risk indicator:	Document metadata contains incorrect dates and/or versions.

Risk case 3:	**Personnel absence due to diseases or other obstacles.**
Risk:	Possible.
Effect:	Moderate or potentially high.
Measure:	Restructure work distribution.
Risk indicator:	Group members stop attending seminars and/or stop responding on the specified communication channel(s).

Risk case 4:	**The client modifies the product requirements.**
Risk:	Low.
Effect:	Moderate–depends on how large the modification is.
Measure:	Revise the product plan.
Risk indicator:	-