

**PROJECT PLAN**

**omniBill-Project**

**Invoicing System**

**omniSpectrum**

**Developing Team**

**HAAGA-HELIA**

Version 0.4 Proposal

Created by Alex Shkunov, Daniel Freitas, Niko Karppinen 30.01.2014

Approved by Steering Group xx.01.2014

**CONTENT**

1 PROJECT DEFINITION 3

1.1 Background 3

1.2 Task 3

1.3 Deliverables 4

1.4 Scope 4

1.5 Risk Analysis 4

1.6 Organization 5

1.6.1 Roles and responsibilities: 5

1.7 Environment 6

1.7.1 Interest Groups of Project Result 6

1.7.2 Construction Environment 6

1.8 Purchases and other costs 6

2 WORK PLAN 7

2.1 Phases 7

2.2 Tasks, workloads and deliverables 7

2.3 Timing 7

2.4 Working Methods and Standards 7

2.5 Project Management 8

3 QUALITY PLAN 9

3.1 Quality Goals 9

3.2 Quality Procedures and Responsibilities 9

3.3 Documentation- and Version Management 9

4 SOURCES AND REFERENCES 10

# 1 PROJECT DEFINITION

## 1.1 Background

Project omniBill was an initiative originally conceived by Niko Karppinen who, after intensive research, decided to come up with a simple solution to invoicing systems. Consequently, this lead to the creation of omniSpectrum, an organization team headed by Alexander Shkunov, Daniel Freitas and Niko Karppinen.

Project omniBill is a simple stand-alone, lightweight and modular invoicing desktop software. Primarily, it will be created for small and micro businesses as well as for individual users. Many small firms still use excel spreadsheets for those purposes, and even though spreadsheets are an universal tool, they were not developed with that purpose in mind, for this reason creating and editing invoices becomes a manual, time consuming and ineffective daily routine. Although it is possible to find some web applications for those purposes, there is a catch - web apps always require hosting which always requires money, so sooner or later the user will need to pay for the service. However, omniBill will be shipped for free. Furthermore, with web apps one needs a persistent Internet connection, with modern technologies that is, unfortunately, not always the case. Additionally, security and privacy of your data in a web application are always a concern.

The omniBill offers an easy and fast way to create, edit and print invoices in conjunction with the possibility to store basic information about customers. Elementary procedures with just a couple of clicks will give the user the needed outcome. At any point in time, it will be possible to check an invoice journal to see the history of previous actions and, for example, print or convert an old invoice into a PDF file.

The goal of the omniBill program is to improve the quality of small business companies and inviduals and thus essentially increasing efficiency and productivity in their work by making their invoicing process faster, more accurate and easier.

## 1.2 Task

The task of the omniBill project is to define, design and eventually build a prototype of the Invoicing System. The prototype is used to test the functionality, usability and the features of the new system, which will be completed and finalized in the follow up construction project during the summer of 2014.

## 1.3 Deliverables

Project deliverables are:

* System Definition and Specification Documentation
* System Design Documentation
* Test plan
* Database and a prototype of the system

Based on the System Requirements Analysis results, project group defines designs and implements the application and related database solution. The omniBill Invoicing System must be compatible with as many platforms as possible. System prototype and related documentation will be delivered to the professor for further evaluation. Quality of all deliverables must meet the criteria described in Chapter 3.1.

Learning objectives for the project team of the omniBill are:

* clear understanding of the development process of information systems
* positive experience of implementing a successful system project
* improved skills in use of object oriented methods in system development process
* applying earlier learned skills in producing the system prototype
* creating and maintaining a positive, productive and constructive group work spirit in project team.

## 1.4 Scope

Project does only the construction of the prototype of omniBill Invoicing system. The implementation of the finalized production version and deployment of the system are not included into this project.

## 1.5 Risk Analysis

Project risks are described and analyzed in Appendix 1.

## 1.6 Organization

Steering Group

(Pekka Kamaja )

Markku Tarkki, chairman

Ulla Vanhanen

Seija Wolfer

**Project Manager**

Niko Karppinen

Project Secretary

(assigned by project team)

**Project Team**

Alex Shkunov

Daniel Freitas

Picture 1. Project Organization

### 1.6.1 Roles and responsibilities:

Niko Karppinen - Project Manager / Requirements Analyst / GUI Designer

Daniel Freitas - System Designer / DB developer / QA / SW Developer

Alex Shkunov - Solution Designer / Software Architect / SW Developer

## 1.7 Environment

### 1.7.1 Interest Groups of Project Result

Team omniSpectrum is the developer and only distributor of the omniBill software for the time being. omniBill is mainly targeted at small business and individuals who are looking for a simple Invoicing System as a solution to handle invoicing. It is a stepping stone for businesses that use Microsoft Excel / Microsoft Word for doing their invoicing.

Theoretically, omniBill should be able to adapt to any environment which makes it the perfect tool. Team omniSpectrum also plans to make the project completely open-source under the BSD license once it is released to the public. For this reason, the team strongly believes that omniBill will be able to become more prominent in the market as a valid solution for expensive proprietary software released by the competition.

### 1.7.2 Construction Environment

Development of software:

1. Systems design and documentation

* MS-Office
* Acrobat Reader
* Google Docs
* Sparx Enterprice Architect
* ArgoUml

1. Creating the prototype and testing

* GitHub git client
* Visual Studio 2012
* SQL Server Express
* Qt Creator
* SQLite library

## 1.8 Purchases and other costs

Project doesn’t require any purchases. Because of the nature of the student project, costs are not allocated for the project. So, only the use of working hours of project team is to be recorded and monitored.

# 2 WORK PLAN

## 2.1 Phases

The project contains managerial and system development tasks. The project is to be monitored by the steering group. The planned milestones are defined in Appendix 2, Project work structure and phases. System Prototype development process will be carried out using component based software engineering approach. Possible stages are described in Appendix 2, Project work structure and phases.

## 2.2 Tasks, workloads and deliverables

Each and every phase and iteration has predefined starting criteria and deliverables. These will be used in quality assurance and steering. Workloads are estimated and will be monitored in hours. The complete workload is estimated to be 315 hours. The size of the project team is 3 persons. Each person is estimated to carry out a workload of 20 hours a week, including also learning activities: class lessons, readings and listening to guest lecturers. Workloads of tasks are described in Appendix 3, Tasks and timing. Project team is responsible for recording the actual use of working hours on weekly basis.

## 2.3 Timing

The project will be initiated at 22.01.2014 and finished at 15.05.2014.  Detailed schedule is described in Appendix 3, Tasks and timing.

## 2.4 Working Methods and Standards

The project applies the working methods used by HAAGA-HELIA:  Rational Unified Process – modeling method and Unified Modeling Language -notation. As well as engineering practices and approaches described in the book “Software engineering(7th edition)”,  Sommerville Ian

In quality assurance, ISO 9000-3 Quality management and quality assurance standards Part 3: Guidelines for the application of ISO 9001 to the development, supply, installation and maintenance of computer software will be applied.

## 2.5 Project Management

Managerial procedures

- The project is carried out according to the approved project plan

Project reporting

- project monitors, records and reports status as described in project plan

Timing

- project applies approved project plan schedule

Reviews and Approval

- quality assurance reviews and testing will be carried out as planned

- the steering group must approve results and accept changes of project plan

Meetings

- steering group meetings will be held according to the project plan

- project team and tutoring meetings will be held weekly

Informing

- steering group takes care of the informing of the project

- project manager informs steering group and project team

- e-mail and Moodle will be used as communication channels for inside information

# 3 QUALITY PLAN

## 3.1 Quality Goals

All project results must meet defined quality standards. The delivered omniBill-system prototype must meet also all customer defined usability requirements. All results must be reviewed, tested and approved as planned.

To keep up the planned schedule is also extremely important. The status of the project and use of resources will be monitored by steering group, project manager and team members as described in project plan.

## 3.2 Quality Procedures and Responsibilities

Reviews and software testing will be used as quality assurance methods. All system documentation must be reviewed. Reviews and testing activities are scheduled in Appendix 3, Tasks and timing. The testing must be described in detail in the Test Plan produced by the project team during the project. A written memo, including error report, is required from each individual review and testing occasion.  Project manager is responsible for these procedures.

Project team is responsible for recording the actual working hours weekly and report them in steering group meetings so, that a comparison of used and planned workloads is shown periodically and cumulatively from the start of the project; on individual level and project total.

## 3.3 Documentation- and Version Management

The documentation must meet sufficient level of standards of good written communication and system development specific standards like UML.

Documents to be stored and delivered are:

Project documentation

-       Project plan

-       Project status reports

-       Final project report

-       Notices and minutes of meetings of project steering

meetings

System documentation

-       System definition and specification Documentation

(OOA)

-       Design documents (OOD)

-       System prototype code (OOP)

-       Testing documents

Version management must be applied to all project documentation and software deliverables.

# 4 SOURCES AND REFERENCES

ISO/IEC 12207. Information technology - Software life cycle processes.

ISO/IEC 6592. Information technology - Guidelines for the documentation of computer-based application systems.

ISO 9000-3. Quality management and quality assurance standards. Part 3: Guidelines for the application of ISO 9001 to the development, supply, installation and maintenance of computer software (ISO 9000-3:1997)

HAAGA-HELIA’s course documents and guidelines.

Project Plan LahjaPaaja in Moodle

Software Engineering Sommerville Ian (7th edition)

**APPENDIX 1        DEFINITION AND ANALYSIS OF PROJECT RISKS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk | Probability | Effect | Reasons | Protection | Actions and measures of consequences of risks |
| Lack of experience | Low | Low product quality | We don’t have much experience with Qt | Learning, trying | First prototype will be developed with C# and only the we will make Qt version |
| Academic workload | Moderate | Delays | A lot of courses in this semester | Spending weekends | Re-planning the project |
| Product competition | Moderate | Less popularity | There some products on a market already | Making it unique and more niche oriented | Keep improvements |

APPENDIX  2       PROJECT WORK STRUCTURE AND PHASES

**1st Iteration (Preparation):**

Project management:

- Software proposal

- Ideas brainstorming

- market research

- Defining the project scope and Development strategy

Requirements Analysis:

- Sketching conceptual Use cases.

- Sketching conceptual ER diagram

Software Design

- Finding appropriate technology

- Defining Solution Design

- Mock ups of User Interface

**2nd Iteration (Analysis and Design):**

Project management:

- Defining Phases and stages of development

- Creating set of tasks

Requirements Analysis:

- Complete ER diagram

- Complete Use Cases

- Defining components

- System design

- Requirements documentation

Software design:

- Database diagram

- Complete UI mock ups

- Components Architecture

- Class diagram

- Test plan

**3rd Iteration (Prototyping and unit testing)**

Software design:

- Individual components complete design

- Assurance of solution integrity

Implementation:

- Physical database implementation

- Basic UI implementation

- Generating Models and implementing data

access layer

- Logical layer component implementation

- Integration and testing

- Adding specific printing and converting

functionalities to components

- Integration and testing

**4th Iteration (Implementation and Quality Assurance)**

Implementation:

- Porting to Qt

Project planning:

- Creating reports

Analysis and design:

- Updating documentation

Quality Assurance:

- Integration and testing

- Creating test reports

APPENDIX  3       TASKS AND TIMING

To be finished later