**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 16/10/2014 | 1.0 | Initialize documentation | Burak Atalay, Bahadır Kırdan, Omer Emre Aslan, Vidal Hara,  Yunus Yılmaz |
| 17/10/2014 | 1.1 | Added Security Risks Analysis and Planning, Configuration Info | Yunus Yılmaz |
| 26/10/2014 | 2.0 | Revised the document | Burak Atalay, Bahadır Kırdan, Omer Emre Aslan, Vidal Hara,  Yunus Yılmaz |

**TABLE OF CONTENTS**

[**Revision Histo**](#h.30j0zll)**r**[**y**](#h.1fob9te)

[**1** **Identificati**](#h.1fob9te)**o**[**n**](#h.2x95bfan0vtk)

[***1.1*** ***Docume***](#h.2x95bfan0vtk)***n***[***t overview***](#h.2et92p0)

[***1***](#h.2et92p0)***.***[***2*** ***Abbrevia***](#h.n80bkwadacdk)***t***[***ions***](#h.3dy6vkm)

[1.2.1 Abbrev](#h.3dy6vkm)i[ations](#h.4d34og8)

[***1.3*** ***References***](#h.4d34og8)

[1.3.1 P](#h.4d34og8)r[oject References](#h.5vn0bddcuibe)

[**2** **Software**](#h.5vn0bddcuibe)[**Development Activities**](#h.hgoqrg3ae73r)

[***2.***](#h.hgoqrg3ae73r)***1*** [***Software development p***](#h.k40liv1ght9)***r***[***ocess***](#h.clllcgykc4f0)

[2.1.1](#h.clllcgykc4f0) O[verview of process phases](#h.9rmqnh9dt4x)

[2.1.2 Techn](#h.35nkun2)i[cal documentation](#h.r1gim2l9jvig)

[2.1.3 Deliverables:](#h.r1gim2l9jvig)

[***2***](#h.r1gim2l9jvig)***.***[***2*** ***Software deve***](#h.lh4k711hyld5)***l***[***opment tools***](#h.9kyzdsjcg3h7)

[2.2.1 Workst](#h.9kyzdsjcg3h7)a[tion](#h.knxs6y25rfnw)

[2.2.2 Requirements](#h.knxs6y25rfnw) [management and documentation](#h.xso6yeimeggf)

[2.2.3 Soft](#h.xso6yeimeggf)w[are Design](#h.2hmi0ip0xerh)

[2.2.](#h.2hmi0ip0xerh)4 [Coding and automated tests](#h.4i7ojhp)

[2.](#h.4i7ojhp)2[.5 Configuratio](#h.cvy88laiatm1)n [management](#h.1ci93xb)

***2***[***.3*** ***Software d***](#h.xrfgos6d0tje)***evelopment rules and standards***

**3** **Responsibilities**

***3.1*** ***Activities and responsibilities***

**4** **Risk Assessment**

***4.1*** ***Risk Analysis***

***4.2*** ***Risk Planning***

# **Identification**

## ***Document overview***

This document contains the software development plan of software OzU Scheduler. OzU Scheduler is a web-based tool that will be developed to help the students of Ozyegin University in choosing and planning the courses that will take.

* Software Development Process – explains development schedule, defines the major phases, technical documentations and deliverables.
* Software Development Tools – explains the workstations to development, management and documentation requirements, software design tools, coding & testing tools and management tools for configurations.
* Activities and Responsibilities – describes the organizational structure of the team.
* Risk Analysis & Planning – describes the estimated risks, risk analysis and the possible solution strategies as suitable precautions.

## ***Abbreviations***

### **Abbreviations**

Play Framework: PF

OzU Scheduler : OzUSch

MVC : Model View Controller

UI: User Interface

DB: Database

SRS: Software Requirements Specification

SDP: Software Development Plan

STP: Software Test Plan

SDD: Software Design Document

STR: Software Test Report

## ***Referenc***es

### **Project References**

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| [R1] | 1 | (March 21, 2014) Google Java Style. Retrieved from: <https://google-styleguide.googlecode.com/svn/trunk/javaguide.html> |

# **Software Development Activities**

The section lists and describes the software development activities of OzUSch software development project.

## ***Software development process***

The software development process chosen for the project is the waterfall process model.

The waterfall process model was chosen for the reasons below:

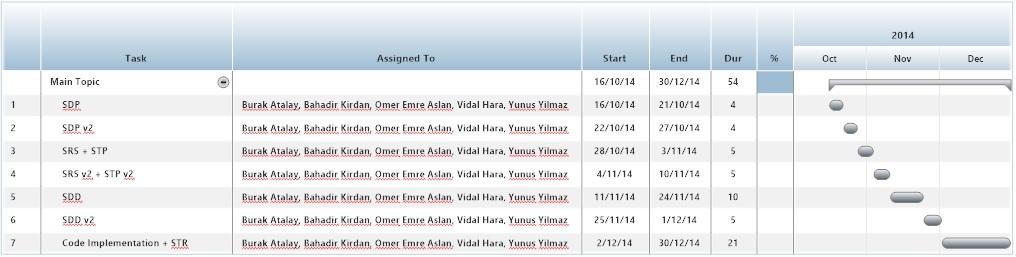
* Since OzuSch is a stable project and it’s requirements are known, the specifications can be determined in the beginning of the project.
* Due to the fact that the project will be developed by multiple people, it will be easier to report and identify what will be done by each individual that will work on the project.

### **Overview of process phases**

The life cycle of the software development project is composed of:

* Software specification,
* Software detailed design,
* Software coding and unit tests,
* Software verification tests

### In the following chart, we depict the schedule of our project.



### **Technical documentation**

The following documentation is produced during the design phases:

* Software specification: SRS, STP,
* Software detailed conception: updated SRS, SDD, updated STP, STD
* Coding and unit tests: STR of unit tests
* Software tests phases : STR

### **Deliverables:**

The following items are delivered both during and at the end of the process:

* OzuSch User Manual: User guideline for selection of courses.
* GitHub Issues: Support service for entering issues/bugs
* GitHub Repository: Support service for updating project

## ***Software development tools***

### **Workstation**

* ACER - Intel Core i5-2450M CPU @ 2.50GHz 2.50GHz, 8 GB RAM, Windows 8.1 OS
* MSI - Intel Core i7-4710HQ CPU @ 2.50GHz (8 CPUs), 16 GB RAM, Windows 8.1 OS
* Apple - Intel Core i5@ 2.60GHz (2 CPUs), 8 GB RAM, OS X Yosemite
* Apple - Intel Core i7@ 2.60GHz (8 CPUs), 8 GB RAM, OS X Yosemite
* Apple - Intel Core i7@ 2.60GHz (8 CPUs), 8 GB RAM, OS X Mavericks
  + 1. **Target System**

All OSwith compatible with Java

### **Requirements management and documentation**

SmartDraw: Used when preparing Gantt charts ([SmartDraw Home Page](http://www.smartdraw.com/))

Github, Dropbox, Google Drive: Used as a storage and online work

MS Word (2010 & 2013): Used for preparing documents

Gmail, Skype, Whatsapp: Used for communication

### Software Design

ArgoUML is an UML diagramming application written in Java and released under the open source Eclipse Public License. By virtue of being a Java application, it is available on any platform supported by Java.

### **Coding and automated tests**

* Eclipse EE

### **Configuration management**

* GitHub: Revision control based storage ([GitHub Project Page](https://github.com/ynsy/CS320)) Hierarchical folder system used (I.e. Documents and Implementation separated).
* GitHub Issue Tracker: Web-based general-purpose issue-tracker. Used for testing and keeping track of the status of bugs/issues

## ***Software development rules and standards***

In this project, Google Java Style**1** Standards is used. Also MVC design pattern will be used

# **Responsibilities**

## ***Activities and responsibilities***

Each activity has someone responsible, mandatory.

|  |  |
| --- | --- |
| **Activity** | **Responsibility** |
| Project Management | Burak Atalay |
| Security Management | Yunus Yılmaz |
| UI Design | Ömer Aslan |
| Debug and Test Management | Vidal Hara, Yunus Yılmaz |
| Database Design | Bahadır Kırdan |
| Coding Implementation | Burak Atalay, Yunus Yılmaz, Ömer Aslan, Vidal Hara, Bahadır Kırdan |

# **Risk Assessment**

## ***Risk Analysis***

|  |  |
| --- | --- |
| Risk Type | Possible Risks |
| Technology | Since everyone is not comfortable with PF, the coding process might take longer than the initial plan |
| Security | We will be using DB and we will be taking inputs from the users. Users can take all DB info it must be controlled. Also, we will create accounts, these info can be reached from hackers. These must be encrypted with more secure crypto algorithm. |
| Requirements | Our source may not provide necessary data |

## ***Risk Planning***

Technology:

* In the beginning of the project, everyone must learn PF.

Requirements:

* The data can be written manually and can be parsed from the Excel file that is given to the students by the school via an automatic parser.

Security:

* Select more secure algorithm and implement to our project. Selected algorithm may be depreciated when we finish the implementation by the hackers.