REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 02/10/2018 | 1.0 | Initial edits on the paper. | Arman Hasanzade,  Kaan Kırlı,  Kaan Yıldız,  Güneş Büyükgönenç |
| 03/10/2018 | 1.1 | Document overview second edit on the paper and workstation part was filled | Yazan Shehab,  Kaan Kırlı |
| 04/10/2018 | 1.2 | Added Computer specs and fixed the headline and removed the wrong items on the Documentation tools part | Arman Hasanzade |
| 06/10/2018 | 1.3 | Added Project References (1.3), Activities and Responsibilities (3.1), Risk Analysis (4.1) and Risk Planning (4.2). | Arman Hasanzade,  Güneş Büyükgönenç,  Kaan Kırlı,  Kaan Yıldız |
| 07/10/2018 | 1.4 | Added Gantt Chart on 2.1.1 Overview of process phases | Arman Hasanzade |
| 08/10/2018 | 2.0 | Overview and finalization of the paper. As well as fixes and some grammatical corrections. Edited Activities & responsibilities (3.1) Added to Risk analysis (4.1) and risk planning (4.2) | Yazan Shehab |
| 15/10/2018 | 2.1 | Edited 2.1.3, 2.2.2, 4.1 and 4.2 | Yazan Shehab, Kaan Yıldız, Kaan Kırlı, Güneş Büyükgönenç |
| 16/10/2018 | 2.2 | Changed the version format.  Fixed the numbers on Table of Contents  Redone the Gantt Chart  Edited 1.2.1, 2.2.2, 2.2.4, edited 4.1 and added Risk Indicator to 4.2 | Arman Hasanzde |

**TABLE OF CONTENTS**

[**1 Identification 3**](#_Toc527549429)

[**1.1 Document overview 3**](#_Toc527549430)

[**1.2 Abbreviations 3**](#_Toc527549431)

[**1.2.1 Abbreviations 3**](#_Toc527549432)

[**1.3 References 3**](#_Toc527549433)

[**1.3.1 Project References 3**](#_Toc527549434)

[**2 Software Development Activities 3**](#_Toc527549435)

[**2.1 Software development process 3**](#_Toc527549436)

[**2.1.1 Overview of process phases 4**](#_Toc527549437)

[**2.1.2 Technical documentation 4**](#_Toc527549438)

[**2.1.3 Deliverables 5**](#_Toc527549439)

[**2.2 Software development tools 5**](#_Toc527549440)

[**2.2.1 Workstation 5**](#_Toc527549441)

[**2.2.2 Requirements management and documentation 6**](#_Toc527549442)

[**2.2.3 Software Design 6**](#_Toc527549443)

[**2.2.4 Coding and automated tests 6**](#_Toc527549444)

[**2.2.5 Configuration management 6**](#_Toc527549445)

[**2.3 Software development rules and standards 6**](#_Toc527549446)

[**3 Responsibilities 6**](#_Toc527549447)

[**3.1 Activities and responsibilities 6**](#_Toc527549448)

[**4 Risk Assessment 7**](#_Toc527549449)

[**4.1 Risk Analysis 7**](#_Toc527549450)

[**4.2 Risk Planning 7**](#_Toc527549451)

# Identification

## Document overview

This document contains the software development plan of software SUP.

Super Paint is a Desktop app which lets the user draw anything they want using various brushes and colors. SUP provides some preset brushes for the user to work with, the user can create their own brushes and save them for future usage anytime they want. The user can also save their drawn pictures to their own computer or open an existing one to work on it.

## Abbreviations

### Abbreviations

SUP: Super Paint

UML: Unified Modeling Language

IDE: Integrated Development Environment

JDK: Java Development Kit

SRS: Software Requirement Specification

STP: Software Test Plan

SDD: Software Design Document

STR: Software Test Report

## References

### Project References

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| [R1] | MSPaint3D | <https://www.microsoft.com/tr-tr/p/paint-3d/9nblggh5fv99#activetab=pivot:overviewtab> |
| [R2] | Gimp | <https://www.gimp.org/> |

# Software Development Activities

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

## Software development process

This is a course project, which adopts the waterfall model as the software development process.

### Overview of process phases

The software development process for the project will be composed of the following phases:

* Planning
* Requirements Analysis
* Design
* Implementation
* Testing

These phases will follow each other sequentially, where each phase starts just after the completion of the previous one. The following Gantt chart depicts the planned start date and duration for the phases.

### Technical documentation

The following documentation is produced during the software development phases:

* Software specification: SRS, STP
* Software detailed conception: SDD
* Software tests phases: STR

### Deliverables

The following items will be delivered at the end of the process:

* Technical documentation as outlined in Section 2.1.2
* Software and its configuration files

* <https://github.com/kaankirli/sup>

## Software development tools

### Workstation

Kaan Kırlı:

Device name: HP Pavilion 14 Notebook

Processor: Intel(R) Core (™) i5-7200U @ 2.50 GHz 2.71 GHz

RAM: 8.00 GB

System Type: 64-bit Operating System, x64 based processor

Operating System: Windows 10 Home Single Language

Arman Hasanzade:

Device name: Asus N552VW Laptop

Processor: Intel(R) Core (™) i7-6700HQ CPU @ 2.60 GHz 2.59 GHz

RAM: 16.00 GB

System Type: 64-bit Operating System, x64 based processor

Operating System: Windows 10 Home Single Language

Kaan Yıldız:

Device name: Apple Macbook Pro 15" Late 2016 Laptop

Processor: Intel(R) Core (™) i7-6700HQ CPU @ 2.60 GHz-3.50 GHz

RAM: 16.00 GB

System Type: 64-bit Operating System, x64 based processor

Operating System: MacOS Mojave (10.14)

Güneş Büyükgönenç:

Device name: Custom Built PC

Processor: Intel(R) Core (™) i5-3450 CPU @ 3.10 GHz-3.10 GHz

RAM: 8.00 GB

System Type: 64-bit Operating System, x64 based processor

Operating System: Windows 10 Education

Yazan Shehab:

Device name: Toshiba tecra

Processor: Intel(R) core (™) i5-5200U CPU @ 2.20GHz 2.20 GHz

RAM: 8.00 GB

System Type: 64-bit Operating System x64-based processor

Operating System: Windows 10 Pro

### Requirements management and documentation

Google Docs, Microsoft Word, Microsoft Excel, GitHub for documentation

Trello for task tracking and team collaboration

### Software Design

Argo UML open source tool

### Coding and automated tests

Eclipse, IntelliJ Idea, JDK, JVM

JUnit for automated testing

MySQL for database

### Configuration management

GitHub[[1]](#footnote-1) will be used for software configuration management and tracking issues regarding the software development. A public repository will be created for this purpose.

## Software development rules and standards

UML[[2]](#footnote-2) will be used for software design documentation.

Java clean coding standards - <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

# Responsibilities

## Activities and responsibilities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Responsibility** | **Comment** |
| Project management | Kaan Kırlı | Creating efficient project plans. |
| Configuration tools management | Kaan Kırlı,  Kaan Yıldız | Creating integrated tools. |
| Setting up the Development tools | Yazan Shehab | Developing the code and detailing the necessary additional tools. |
| Software specifications | Yazan Shehab | Adding specified user demands and coding the required needs. |
| Database design | Kaan Yıldız,  Arman Hasanzade,  Kaan Kırlı | Creating a manageable and scalable database. |
| Implementation | Güneş Büyükgönenç,  Arman Hasanzade,  Kaan Yıldız | Implementing the design. |
| Testing | Arman Hasanzade  Yazan Shehab | Performing Unit and Integration tests. |

# Risk Assessment

## Risk Analysis

|  |  |  |
| --- | --- | --- |
| **Risks** | **Probability** | **Effect** |
| The time required to develop the software is underestimated. | High | Serious |
| Software Tools cannot be integrated. | Moderate | Tolerable |
| Organization of the team is not efficient. | Moderate | Tolerable |
| Withdrawal of team members. | Moderate | Tolerable |
| Insufficient skill level of members. | Low | Tolerable |

## Risk Planning

|  |  |  |
| --- | --- | --- |
| **Risk** | **Indicators** | **Strategies** |
| The time required to develop the software is underestimated. | Failure to meet the agreed schedule. | Split the roles so that focus would be applied to the underdeveloped segments. |
| Software Tools cannot be integrated. | Having problems with database connection, having problems with GUI and model connections. | Changing software such that it would be integrated with the other tools. |
| Organization of the team is not efficient. | Inability of the team following the dates on the Gantt Chart. | The problem could be solved by scheduling some meetings each group member would discuss the problem and solve it. |
| Withdrawal of team members. | Poor student morale, poor relationship amongst the team members. | Rearranging roles of the group members such that the members could fill the ex-group member’s responsibilities. |
| Insufficient skill level of members. | Not being able to continue the development, not being able to fix the occurring problems | Grouping team members such that the members with low skill level would work with the members with high skill level. |

1. http://www.github.com [↑](#footnote-ref-1)
2. http://www.uml.org/ [↑](#footnote-ref-2)