

CTIS411 Senior Project I

Software Project Management Plan

Left-Over!

(T)hank (G)od (E)ast (O)ver

Furkan Kılıç, 21601729

Mehmet Melikşah Özceyhan, 21703661

Nevin Didem Bilgihan, 21601723

Yağmur Cansu Gürkan, 21601486

Bilkent University

Department of Information Systems and Technologies

27.12.2021

Change History

File Name	Document Type	Deliverable	Version	Submission Date
Deliverable_3_v0.1	Google Docs	3	0.1	-
Deliverable_3_v0.2	Google Docs	3	0.2	-
Deliverable_3_v0.3	Google Docs	3	0.3	-
TGEO_Deliverable_3_1	PDF	3	1	6.12.2021
TGEO_Deliverable_3_2	PDF	3	2	27.12.2021

Project Team

Project Team Name		(T)hank (G)od (E)ast (O)ver	
Name, Surname	Student Id Number	Department	e-mail
Nevin Didem Bilgihan	21601723	CTIS	didem.bilgihan@bilkent.edu.tr
Mehmet Melikşah Özceyhan	21703661	CTIS	meliksah.ozceyhan@ug.bilkent.edu.tr
Furkan Kılıç	21601729	CTIS	furkan.kilic@ug.bilkent.edu.tr
Yağmur Cansu Gürkan	21601486	CTIS	yagmur91598@gmail.com

Project Details

Project Name	Left-Over!
Academic Advisor	Neşe Şahin Özçelik
Github URL	https://github.com/meliksahozceyhan/Senior-Project-LeftOver
WEB page	-

Executive Summary

This document gives detailed information about project management. It includes estimations based on work breakdown structure, use-cases and user stories. As a result of the calculations, the total effort estimation is 2240.784 person hours based on use-case points. Thus, it is an adequate value for the student senior project. In addition, the project schedule is explained with work packages in different phases of the project, resource allocation, dependencies, and budget requirements. After work packages and sprints are planned, the estimated release date is determined as 13th of May 2022. Also, how the project will be tracked by using Trello based on work packages, and how work package completion dates will be used to monitor team performance is explained. Moreover, verification and validation tools and techniques are determined, Amazon Web Services device farm and user manual test approaches will be used. Additionally, various development environments are shown by table, such as typescript and Dart as programming languages, also NestJS and Flutter frameworks will be used for development purposes. Also, the discussion part details the impact of the software project management plan documents in many aspects.

Changelog

- Team table is reformatted.
- Executive summary is fixed by corrected values and evaluation of the estimation is added.
- URL acronym is corrected, and MS is removed from Abbreviations.
- In the scope the version number is mentioned for SRS document.
- WBS is edited, the number of days is added to the table (based on bottom-up approach), the outputs are mentioned in the diagram. Also, the development part is divided by colors to show the output of each increment.
- “Personal Hours” is changed with “person hours”
- Effort estimation is recalculated with person hours multiplier 28.
- \$ sign usage is fixed.
- Time metric is redefined and explained
- Estimation determination and explanation is written
- Development Environment Table reorganized for multiple pages
- References are rewritten

Table of Contents

	<u>Page Number</u>
1. Scope	1
2. Project Effort Estimation	2
3. Project Schedule	9
4. Project Monitoring and Measuring	12
5. Product Verification and Validation	13
6. Software Development Environment	14
7. Discussions	17
1. Limitations and Constraints	17
2. Applicability of the project under real life situations	17
3. Health and Safety Issues	17
4. Legal Issues	17
5. Economic Issues and Constraints	18
6. Sustainability	18
7. Producibility-Manufacturability	18
8. Social, Political and Ethical Issues	18
9. Multidisciplinary Collaboration	19
10. Environmental Issues	19
8. References	20

List of Tables

	<u>Pages</u>
<i>Table 1: Unadjusted Actor Weight</i>	5
<i>Table 2: Unadjusted Use-Case Weight</i>	5
<i>Table 3: Technical Factor Value</i>	6
<i>Table 4: Environmental Factor Value</i>	7
<i>Table 5: Agile Estimation</i>	8
<i>Table 6: Software Development Environment Description</i>	14

List of Figures

Pages

Figure 1: WBS Part 1 2

Figure 2: WBS Part 2 3

Figure 3: WBS Part 3 4

Abbreviations

API	Application Programming Interface
AWS	Amazon Web Services
CTIS	Information Systems and Technologies
EF	Environmental Factor
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
SDD	Software Design Description
SPMP	Software Project Management Plan
SRS	Software Requirements Specification
TCF	Technical Complexity Factor
TF	Technical Factor
UAW	Unadjusted Actor Weight
UC	Use Case
UCP	Adjusted Use Case Points
UI	User Interface
UUCP	Unadjusted Use Case Points
UUCW	Unadjusted Use-Case Weight
URL	Uniform Resource Locator
US	User Story

1. Scope

This section of the document will briefly mention the planned activities that are going to be done by the team during the preparation of this document. The following parts of the document covers work packages, schedule, cost, and effort estimations of the project. Also, the results of the calculations and availability of team members are considered for defining milestones of product deliveries. In this way, an increase in the efficiency and productivity of the team is aimed.

The Software Project Management Plan (SPMP) document is prepared based on requirements determined in the second version of Software Requirements Specification (SRS) document. Moreover, the use cases, which are used to calculate effort estimation and create user stories, are not redefined in this document, they were all predefined in SRS. User stories are simple descriptions of user roles in each use case. Therefore, the user story points are assigned depending on the use case's complexity. As a result, specific time intervals are specified for milestones, which reveals the release date of the product and the completion date of the project.

2. Project Effort Estimation

The following sections indicate the work breakdown structure, use case based, and agile estimation techniques.

1. Work Breakdown Structure (Decomposition-Based) Estimation:

Below diagrams illustrate the work breakdown structure of the Left-Over! System.

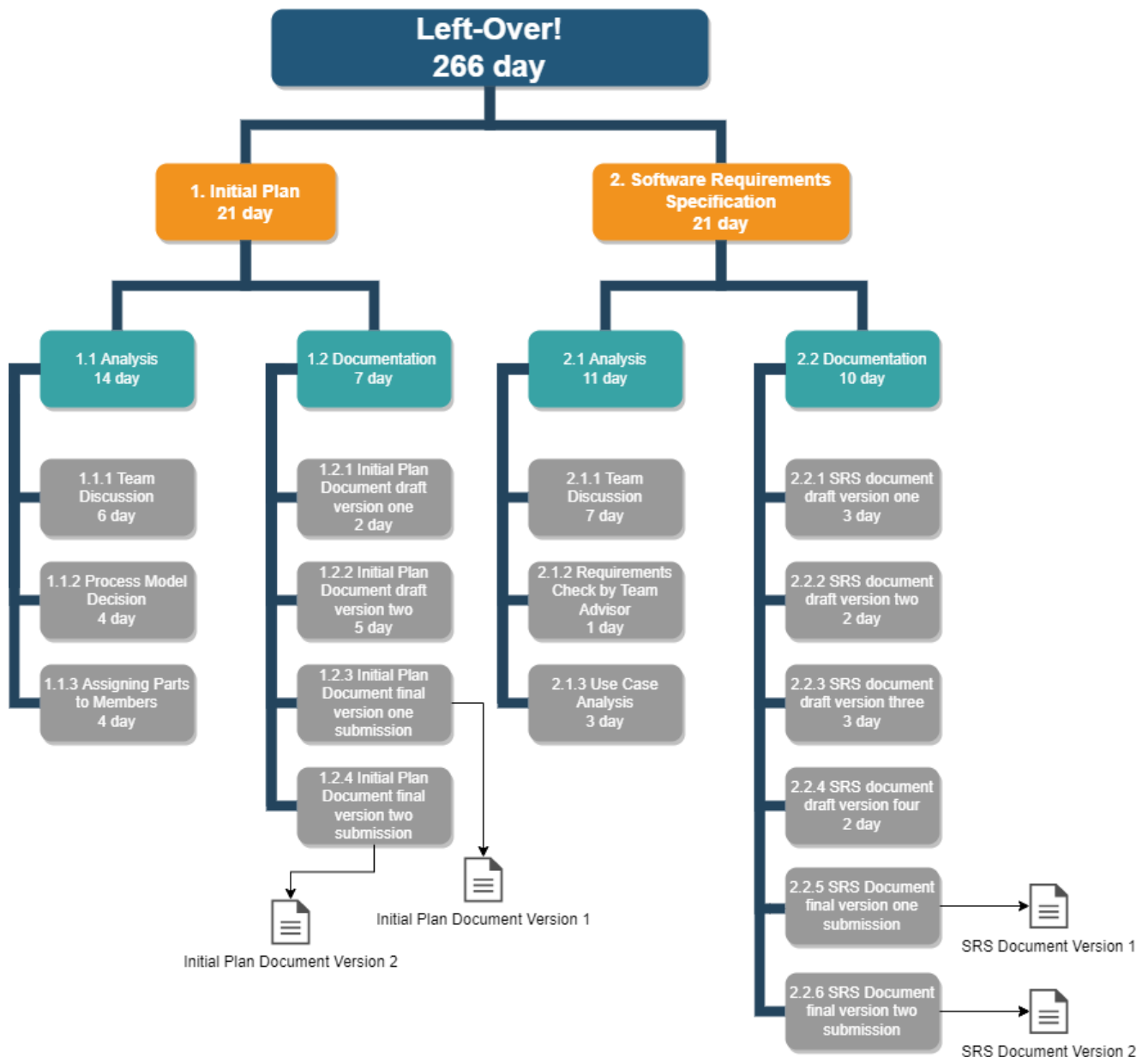


Figure 1: WBS Part 1

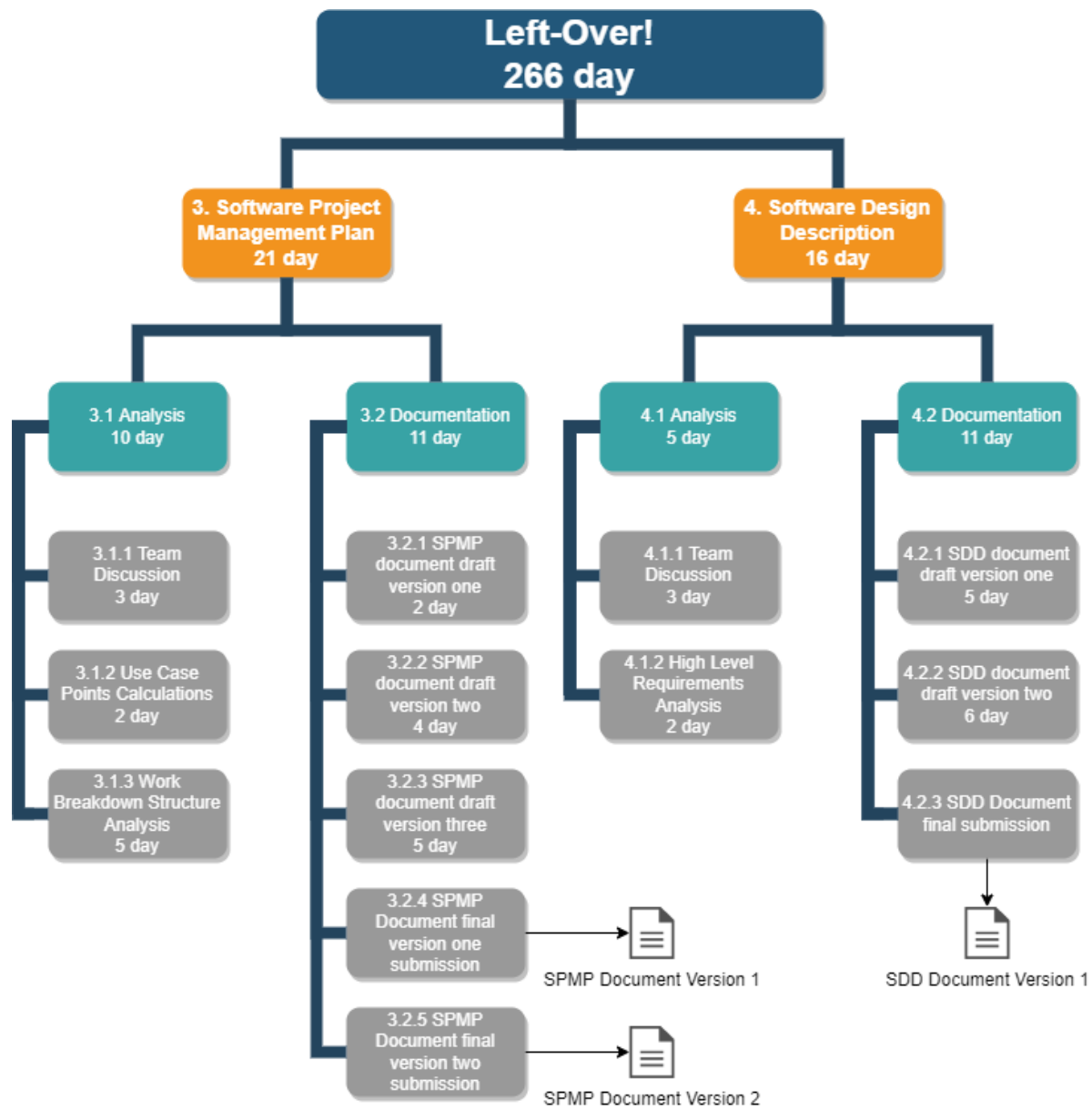


Figure 2: WBS Part 2

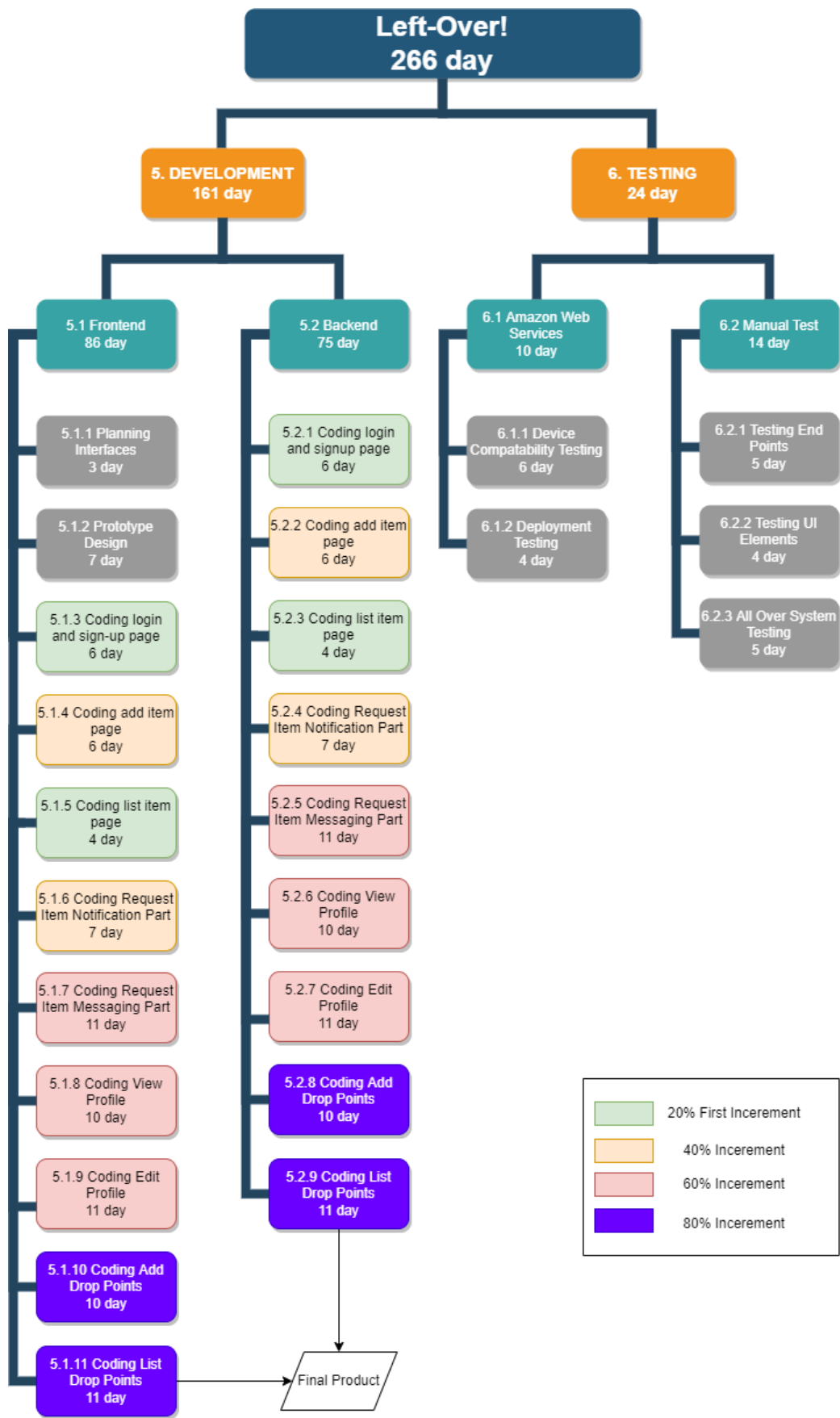


Figure 3: WBS Part 3

2. Use Case Based Estimation:

In this section, tables indicate the use case points and their calculations.

Table 1: Unadjusted Actor Weight

Actor Type	Weighting Factor	Number	Result
Simple	1	4	4
Average	2	0	0
Complex	3	3	9
<i>Unadjusted Actor Weight Total (UAW)</i>			13

Table 2: Unadjusted Use-Case Weight

Use Case Type	Weighting Factor	Number	Result
Simple	5	5	25
Average	10	1	10
Complex	15	4	60
<i>Unadjusted Use-Case Weight (UUCW) Total</i>			95

Table 3: Technical Factor Value

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T1	Distributed system	2.0	0	0
T2	Response time or throughput performance objectives	1.0	4	4
T3	End-user online efficiency	1.0	4	4
T4	Complex internal processing	1.0	1	1
T5	Reusability of code	1.0	1	1
T6	Ease of installation	0.5	3	1.5
T7	Ease of use	0.5	5	2.5
T8	Portability	2.0	2	4
T9	Ease of change	1.0	0	0
T10	Concurrency	1.0	0	0
T11	Special security objectives included	1.0	0	0
T12	Direct access for third parties	1.0	0	0
T13	Special user training required	1.0	0	0
Technical Factor Value (Tfactor)				18

Technical Complexity Factor (TCF) = $0.6 + (0.01 * Tfactor)$

TCF = $0.6 + (0.01 * 18) = 0.78$

Table 4: Environmental Factor Value

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
E1	Familiarity with system development process being used	1.5	2	3
E2	Application experience	0.5	2	1
E3	Object-oriented experience	1.0	3	3
E4	Lead analyst capability	0.5	0	0
E5	Motivation	1.0	4	4
E6	Requirements stability	2.0	3	6
E7	Part time staff	-1.0	0	0
E8	Difficulty of programming language	-1.0	2	-2
Environmental Factor Value (Efactor)				15

Environmental Factor (EF) = $1.4 + (-0.03 * \text{Efactor})$

EF = $1.4 + (-0.03 * 15) = 0.95$

Adjusted Use Case Points (UCP) = $\text{UUCP} * \text{TCF} * \text{EF}$

UCP = $108 * 0.78 * 0.95 = 80.028$

Effort (in Person Hours) = $80.028 * 28 = 2240.784$

3. Agile Estimation:

The table below shows user stories of use cases and their story points.

Table 5: Agile Estimation

Use Case	User Story	Story Point
UC1 – Sign up	US3 – As a user, I should sign up to Left-Over! To use the application.	2
UC2 – Login	US2 – As a user, I want to login to the system to see my account.	1
UC3 – Add Item	US8 – As a user, I want to add items to donate my unused items.	21
UC4 – List Item	US7 – As a user, I want to list items to find my needed items.	13
UC5 – Request Item – Notification & Message	US9 - As a user, I want to be notified by the system to display the requested transaction.	35
	US10 – As a user, I want to message with a receiver about the requested item or message with a recipient to get the item.	55
UC6 – View Profile	US1 - As a user, I want to view my profile to see my process.	0.5
UC7 – Edit Profile	US6 – As a user, I want to edit my profile to change my properties.	8
UC8 – List Drop Points	US4 – As a user, I want to see the listed drop points to throw away my recyclable items.	3
UC9 – Add Drop Points	US5 - As an admin, I want to add a drop point to list them.	5

Above diagrams show the estimation of the completion time for the project using WBS estimation with the bottom-up approach. According to the planned delivery dates of the work packages, the overall project completion time is estimated. Also, when working days are taken as 8 hours with the output of this estimation, the total completion time of the project is equal to 2,188 hours. After estimations based on the WBS are finished the team moved on to the use case-based estimation. Afterwards analyzing use cases and classifying them according to transactions made when issuing the use cases and actors interacting with the use cases. The team came up with a UUCP value that represents the total points generated by the Use cases. Later on, the team has analyzed the project to define environmental and technical factors that affect the project's overall complexity. These factors have been used to estimate the total project effort and as a result the total value of estimation from use cases appears to be 2240.784 person hours which is in accordance with the WBS based estimations of the project.

3. Project Schedule

In this section, activities, deliverables, and milestones will be listed within the project. Also, planned start and finish date, duration, budget, and resources will be assigned to each activity. The project schedule is available as a Microsoft Project named as TGEO_Gantt.mpp.

1) Work Packages

In the work breakdown structure estimation part of the project effort estimation section, work packages are shown in detail. Also, generated Microsoft Project document explains dependencies between work packages under the view tab, network diagram view.

2) Dependencies

Generated Microsoft Project document's network diagram part specifies the ordering relations among work packages to account for interdependencies among them and dependencies on external events. This part is also available under network diagram view under views tab.

3) Resource Requirements

In order to complete the project, 2240.784 person hours are needed so each team member should work 560 hours. The project needs simultaneous work to be completed in time so every team member will use his/her own computer. Detailed resource allocation is shown on the Microsoft Project document. Detailed working times of resources allocated could be seen under report tab resources and resource overview report.

4) Budget and Resource Allocation

Since this project is a student project, team members are not paid. General cost is the expenditures made to services like Amazon Web Services (AWS) and fees for the application market's developer access. The cost for the developer accounts on the markets is fixed and it is \$25 for Play Store and \$100 for App Store annually. On the other hand, the AWS services are priced according to usage; it is expected to be between five to ten dollars per month.

5) Schedule

This subsection provides the schedules of the project functions, activities, and tasks based on the required milestones. A detailed schedule and Gantt chart can be accessed via the Microsoft Project document uploaded with this document.

4. Project Monitoring and Measuring

To keep track of the backlogs of the project workload and work packages, the team will use the Trello tool. It will help the team to create post-it like cards for defining work packages which will be done in each sprint. After each sprint team will come together to decide what to do and how to do in the next sprint. Trello has a concept called boards which will enable the team to specify the work packages to be grouped under different heading. Furthermore, according to the management plan, scrum meetings are arranged face-to-face for the first semester, but for the second semester the meetings will be held via Zoom.

The metrics which are going to be stored are:

- The time difference between planned Work Package completion date and actual Work Package commit date.
- Team members commit dates and how often they commit changes to the source code.

Reasons for collecting these metrics and how these will be collected is explained below.

The team will use Trello's feature which keeps time records of cards' movements between boards. Work package defined cards will be moved across different boards. This moving action is going to define the state of that work-package. Therefore, when a card is moved to a different board all members of the development team will be notified. Thus, team members will be knowledgeable about the track of the project. The team will require this information to keep track of the completed works and also if a member finished their part earlier than others, the member could help other members. Furthermore, the tracked time information will be collected for calculating the efficiency and overall time spent on the project development process. Moreover, data from the git merge requests and git commits will be used to keep track of who completed which part of the project, and how often the team members contribute to the software development process.

5. Product Verification and Validation

Software verification process is an essential part of the project because it provides the evidence that the system or system element performs its intended functions and meets all the expected requirements [1]. In order to find the errors, gaps, or missing requirements in comparison to the actual requirements, various testing tools and techniques will be used in the project.

The four main stages of testing which are unit testing, integration testing, system testing, and acceptance testing will be considered as testing techniques to be used while testing the software. Unit testing will focus on the specific units of the software to determine whether each one is fully functional. Integration testing will allow the team members to combine all the units within a program and test them as a group. System testing will be a complete application test to evaluate whether the system fully satisfies all the expected requirements. Acceptance testing will determine whether the system is ready for release to customers or jury [2].

Although the project software development team will manually test whether the software meets the expected requirements by looking at the product from the end users' perspective, some test automation tools will be used to get more accurate and faster results. AWS device farm will be used to test and qualify the software.

6. Software Development Environment

The following table indicates information details about tools, libraries, programming languages, frameworks, database management system, Application Programming Interfaces (API), document editor and Integrated Development Environment (IDE) that are planned to be used including versions, brief descriptions, and Uniform Resource Locator (URL) of them.

Table 6: Software Development Environment Description

Name	Type	Version	Description	URL
Dart	Programming Language	2.12.4	Dart is a language optimized for fast apps on any platform which empowers flutter.	https://dart.dev
Typescript	Programming Language	4.5	TypeScript is a typed version of javascript which empowers NestJS.	https://www.typescriptlang.org
Flutter	Framework	2.5.3	Flutter will enable creating android and iOS apps from a single codebase.	https://flutter.dev
NestJS	Framework	8	NestJS framework will be used for building efficient, scalable server-side applications which will be used by mobile applications.	https://nestjs.com
PostgreSQL	Database Management System	14	The product will use PostgreSQL as a database management system to store data.	https://www.postgresql.org
Google Maps for Flutter	API	2.1.1	A Flutter plugin that provides a Google Maps widget.	https://pub.dev/packages/google_maps_flutter
Google Docs	Document Editor	1.21.442.01.30	A document editor which enables collaborative works. It is used for creating project documents.	https://docs.google.com/

Name	Type	Version	Description	URL
Microsoft Project	Project Management Tool	Version 2110	Microsoft Project is used to create a gantt chart which lists the tasks in Leftover! project.	https://www.microsoft.com/en-us/microsoft-365/project/project-management-software
Draw.io	Chart and Diagram Tool	13.9.9	Draw.io is used to create class, activity, sequence, etc. diagrams of the system.	https://drawio-app.com
Figma	UI Design Tool	1.0	Figma is a UI/UX design tool for prototyping, design and code-generation.	https://www.figma.com
GitHub	Version Management Tool	2.34.0	Used to keep track of the versions of the project and to enable the project team to work collaboratively on code.	https://github.com
Visual Studio Code	IDE	1.62	Code editor	https://code.visualstudio.com
Moment.JS	Library	2.29.1	Moment.js is a package that will ease the use and modify the date values.	https://momentjs.com/
Socket.IO	Library	4.4.0	This library will provide an easy way to develop a real time messaging platform.	https://socket.io/
Docker	Software	4.2.0	It enables the product to be packaged into a container which is a standardized executable component combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment.	https://www.docker.com/
i18n	Library	0.13.3	i18n will provide multi language support thus it will make the application available in different languages.	https://www.i18next.com

Name	Type	Version	Description	URL
Trello	Collaboration Tool	2.10.8	Trello is a tool to provide an environment to collaborate and manage projects.	https://trello.com

7. Discussions

The discussions mention the validity and applicability of the SPMP document. Furthermore, SPMP's effects are interpreted in social and universal aspects considering their social, environmental, and legal implications.

1. Limitations and Constraints

Limitations and constraints may influence how the team will manage the projects. The limitations can fall into several categories. By recognizing these categories, their effect can be minimized. Scheduling is the crucial limitation that the team must follow. The determined time periods of the software process model and deadlines should be taken into account. The constraint is defining time and budget information on the SPMP document is calculated according to deadlines of projects. Thus, the team may have to work overtime to complete requirements specified in the Software Requirements Specification document.

2. Applicability of the project under real life situations

Under real life situations the SPMP document will need to define the budget and completion time of the project considering customers' needs. Since this is a student project, the team has defined deadlines to complete the work and has no budget. Thus, this document will become less effective for this project compared to the real-life documents.

3. Health and Safety Issues

While writing this document team members can be stressed because of the tight schedule and heavy workload. This excessive stress might cause different effects on team members' psychology. Also, while developing the project team members will use computers and will have to look at screens for long hours so this might cause some sighting issues.

4. Legal Issues

There are no legal issues considering the SPMP. Since this is a student project, team members do not need to obey the Turkish Social Security Administration's working rules in terms of time and budget.

5. Economic Issues and Constraints

During writing the SPMP document, team members must stay late for many days, so some of them had to turn home by taxi. Also, since it was a long document, team members gathered more frequently, and it affected gasoline expenses of team members who have cars. Furthermore, since the team does not have an office to work in, meetings were made outside, which led to an increase in food and coffee expenses. Moreover, since the team does not expect any payment, the management does not have any costs.

6. Sustainability

A common point around school is determined as a meeting point to make SPMP document meetings regularly. Also, since it is a school project and there is no hierarchy among team members, the management is sustained with a common benefit.

7. Producibility-Manufacturability

Left-Over! does not serve a tangible product directly, it offers software. Hence, there will not be continuous production. Therefore, the management plan is determined in a way that once the product is released according to requirements specified in the Software Requirement Specification (SRS) document, it will not need any maintenance and improvements. Hence, the SPMP document will not be touched except for required changes determined in the feedback session. Furthermore, the management plan was set at the beginning stage of the project and considering the emergencies it can be reshaped during the lifetime of the project.

8. Social, Political and Ethical Issues

When preparing this document and the process of software management plan the team distributes the work equally among them to prevent a feel of injustice about the work done by them. Moreover, during the writing of this document, the right to freedom of thought and expression, which is one of the main issues of fundamental human rights, was taken into consideration, all team members expressed their opinions freely and no discrimination was made.

9. Multidisciplinary Collaboration

There is no multidisciplinary collaboration in management since all members of the team are from the Information Systems and Technologies (CTIS) department including developer team and team advisor. Thus, no advice or consultancy is taken from other disciplines.

10. Environmental Issues

Online management tools are used such as Google documents, Trello, and Microsoft Project to prevent paper waste during the project management planning process. However, all team members use their personal computers separately for documentation, which causes more electricity consumption. Therefore, the process planned to finish the project accordingly so that the consumption of electricity to be at minimum.

8. References

- [1] "Verification", *Dau.edu*, 2021. [Online]. Available: <https://www.dau.edu/tools/se-brainbook/Pages/Technical%20Processes/verification.aspx>. [Accessed: 08- Nov- 2021].
- [2] "The Four Levels of Software Testing", *Segue Technologies*, 2021. [Online]. Available: <https://www.seguetech.com/the-four-levels-of-software-testing/>. [Accessed: 08- Nov- 2021].