



**Michigan
Technological
University**

College of Computing
Computer Science Department
CS3141 Team Software Project
Spring 2022

Simplyst

Section: R02

Team #: 9

Roll #	Name	Role
1	Liam Biggers-Brookhouse	Scrum Master
2	Jordan Bramer	Developer
3	Reese Bunker	Developer
4	Grace Chandler	Developer

Instructor:

Serein AL-Ratrout

Abstract

(Abstract is one-paragraph summarizes your project, describes the content and scope of the project objective, methodology, findings, and conclusion. So, you need to write one-paragraph that gives an abstract idea about the entire project, the aim of the project, the process model you used, the tools, what you have done, the results, and your conclusion. If you think the project is worth extending to a Final Year Project (FYP) by you or other students or can be adopted and extended by industry/market, then mention that here and add it also as future work.)

Example of abstract

In this project a student registration web application for university students and staff was developed, the aim of this application is to provide a simple set-up of programs for student enrolment, improve efficiencies and eliminate unnecessary paperwork. The system mainly has two modules: students and staffs. Students can create account then sign in/out, add, update, delete, and modify their data and schedule. Staff can also create account and then sign in, add, update, delete, and modify their data. Waterfall process model followed during project development and the implementation was realized by use of object-oriented PHP, HTML, MYSQL and Dreamweaver technologies.

It has been found that the final system was simple and user friendly with easy user interface, hence the end-users do not need to undergo extensive training or require any special skills. It was also secure, and reliable.

Not in chapter 1, saving for end for end to make it as effective as possible.

Table of Contents

Table of Figures

Chapter1

Specification

1.1 Introduction

Task apps are common tools that list things a person has to do to help remind them. Our web application will do this, while also having the option for recurring tasks so that the user doesn't have to put it into the list every time. It will also have the option to make tasks be entered for specific times. These features will help the users be more productive and organised.

1.2 Problem Statement

Especially with the recent movement to declinicalize psychology, accompanying the normalisation and proliferation of diagnoses like ADHD, more and more people are accepting and even embracing tools that facilitate productivity. A common struggle is accomplishing all the goals a person has during a week. When things need to be repeatedly done, they often get forgotten or procrastinated. We all know the problem of having to wear old or weird clothes when you haven't done laundry, which could be avoided if you just did your laundry earlier. Feeding your fish? Watering your plants? When you forget, they die. Using a to-do list that has recurring events can solve this common problem. Outsource your executive function with this fun, exciting app!

1.3 Aim and Objectives

Aim:

The aim for this project is to make the most simple and effective todo list possible. The app should have useful features, but only so far as they can serve to enhance simplicity for the user.

Objectives:

- To be incredibly low stress to use
- To create markers of real world tasks for the users
- To make interacting with markers upon completion of those tasks feel rewarding
- To be able to set recurring tasks

1.4 Stakeholders

The stakeholders are the entities that have tasked the team with the project. They set requirements, needs for the project, and are financially invested. Additionally, the direct users are stakeholders as well. That is, people who struggle with maintaining a schedule or forget often.

1.5 Methodology

Our team will be using an incremental approach using the scrum methodology. We are currently brainstorming functionality for the app. After the project begins we plan to have weekly meetings discussing what we've done in the past week and what we will do in the next week. Goals will be set by each team member individually.

1.6 Tools

- Using HTML/JS
- Cookies for saving task progress
- Ideally accessible on mobile devices and PCs

1.7 High-Level Business Requirements

★ Functional Requirements

- Adding and deleting tasks.
- Setting recurring tasks.
- Adding dates and times of tasks.
- Displaying daily task list.
- Displaying upcoming tasks.
- Checking off completed tasks.
- Edit tasks.

★ Non-functional requirements

- Very simple and minimalist
- Accessible on lots of devices
- Inexpensive to system resources
- Visually Pleasing

1.8 Product backlog

You can use the following table:

Priority	User Story	Tasks	Estimated effort	Sprint
1	As a User, I want to be able to add, delete, and edit my tasks.	Design and implement the interface	1 H	1
		Adding and Deleting tasks	1 H	
		Edit tasks	1.5 H	
2	As a User, I want to be able to see: tasks with dates and times as well as general, non-time based tasks	Design and implement the interface	2 H	1
2	As a User, I want to check off tasks from my list.	Design and implement interface	1 H	2
		Checking off tasks	1 H	

2	As a User, I want to have categories for tasks	Design and implement interface	1 H	2
		Create categories	1 H	
1	As a User, I want to be able to see my current tasks, as well as upcoming tasks.	Design and implement interface	1 H	1
		View current tasks	1 H	
		View upcoming tasks	2 H	
3	As a User, I want a celebratory animation/sound when I check off a task.	Create animation	1 H	3
		Play animation when task is marked complete	1 H	
3	As a User, I want to be able to choose a theme for my application.	Design and implement interface	1 H	3
		Create themes	1 H	
3	As a User, I want to have collapsible notes for my tasks	Create Notes	1 H	3

Chapter 2

Analysis and Design

Chapter 3

Implementation

The following [report](#) is a good example that you can follow for implementation please refer to pages 25 - 30, and here is another [example](#) for your reference.

Chapter 4

Validation

For Chapter 4 (Validation): here you need to write about the process of checking that your software system meets specifications and requirements so that it fulfils its intended purpose, and to confirm or to prove the accuracy of your project.

Write about your testing and validation; **level of testing** you had, unit testing, integration testing, validation testing and acceptance testing. Did you have **manual or automated** testing or both? specify the part(s) that have automated testing and part(s) that have manual testing, and **What is your oracle?**

Write the test cases for valid and invalid **input** (please see Week3 Automated Testing/slide 11),

then confirm that no errors in the code and the application is able to operate in required condition (OS, web browsers) and you have created the code correctly.

For validation and acceptance testing write who tested your system? MTU students? computer science student? other department students? your group only? other college students? public users? How many students/users? How many times? could they use it easily or did they make mistakes?

Chapter 5

Limitations and Future Work

For Chapter 5

Limitations: address everything that the project left, if some project backlog items/features/ requirements have not been implemented then mention them in this part with an explanation/justification why you couldn't implement them (Time constraints the time was not enough, some developers were unavailable, because of COVID19, or tool limitationetc.). Many students tend to feel that presenting the limits of their work makes work weaker. on the contrary, approaching this section shows maturity for the academic universe, and writing about them actually strengthens your work by identifying any problems before reviewers/readers find them.

Future work : if the limitations can be addressed in the future then add this in here in future work, moreover, if you believe this project can be extendable (add more features/more parts) that the project is worth extending to a Final Year Project (FYP) by you or other students or can be adopted and extended by industry as a product so you can give directions for that in future work.

Chapter 6

Conclusion

For Chapter 6 (Conclusion),: write what you have concluded.

Examples:

I solved many problems in the project...

This application/project/system was applied to improve the learning process.

The results of this project showed that system significantly facilitated the students' learning process.

The system is useless, acceptable, usable, beneficial or maybe enjoyable and why do you believe that.

References

(Include any references to external documents or materials (for example, tutorials the team will be using, literature , web references or links to documentation of third-party tools you will use) here.

The references should be properly numbered and correctly used in the text.

The Reference section should be in the following fashion:

References

Journal, Magazine/ Newspaper Article

[1] Author, "Title," *Journal name*, p. pages, year.

Book

[2] Author, Book Title, publisher, year.

Internet Web page:

[3] Author, "Name of the Web Page," [Online]. Available: URL. [Accessed Date].