Michigan Technological University

College of Computing

Computer Science Department

CS3141 Team Software Project

Spring 2022

# Project Title

Section: R01

Team #: 5

| Roll # | Name | Role |
|---|---|---|
| 24 | Ricardo Nunez Cuesta | Scrum Master |
| 22 | Sean McCarry | Developer |
| 23 | Damion Miller | Developer |
| 21 | Noah Kolczynski | Developer |
| 13 | Kelsey Hagebusch | 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.

_____

_____

_____

_____

_____

_____

# Table of Contents

## Table of Figures

# Chapter1

## Specification

## 1.1 Introduction

This project aims to provide an online portal to make renting more efficient, benefiting both landlords and tenants. We will provide a web app in which the tenant will be charged every month using a payment method of their choosing. With that, we will make sure the landlord gets the money on time, and can easily track which tenants have paid. We will send an email notification to the tenant one day before the payment is made. In case their payment does not go through we will notify them every day. If they do not pay the landlord will have the option to send an eviction notice. The expected result of the project is that the tenant will be more responsible and they will prioritise having the money to pay the monthly rent and we will minimise the interaction between the landlord and the tenant. We will add more functionalities to the app in which the tenant can create a maintenance request.

## 1.2 Problem Statement

What? A portal system for landlords and tenants. Landlords will be able to assign properties to tenants, set rent prices, manage maintenance requests, message tenants, and view who has paid rent. Tenants will be able to pay their bills, submit maintenance requests, and message their landlords.

Why? Many tenants will either not pay their rent or pay late if they need to pay the bill physically, and landlords will want to be able to keep track of which tenants have paid easily.

Tenants also want an easy way to see their bill every month, as well as have easy access to maintenance requests.

We will help the landlord get the tenants payments in time. Facilitate access to tenants. It will also give landlords an easy method to track their properties and have easy access to information about tenants.

## 1.3 Aim and Objectives

The aim of this project is to develop a web application to facilitate the apartment rental process for landlords and tenants. This web application will allow landlords to keep track of their properties, including which apartments are being rented out, which are vacant, and any maintenance requests, set rent for different properties and different apartment types, add tenants to an apartment, view which tenants have paid their rent each month, and send an eviction notice via email. Tenants will be able to view previous and upcoming bills, pay their bills, create maintenance requests, and view the address for their apartment.

Objectives:

- Allow renters to easily view past/upcoming bills
- Allow renters to create maintenance requests
- Display information about occupied properties to landlords
- Allow landlords to view vacant and occupied properties
- Simplify the overall renting process for tenants and landlords

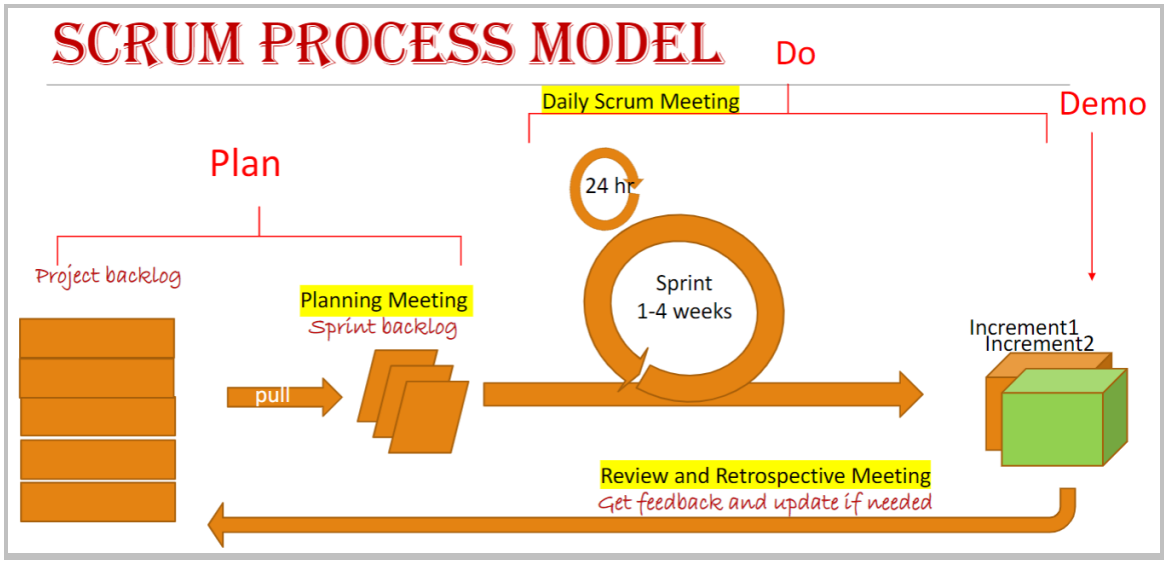
## 1.4 Stakeholders

everyone who is involved using or developing the app
Developers, designer, software engineers, landlords and tenants

## 1.5 Methodology

We will be using the scrum software process. Requirements will be organized into the project backlog and then separated into sprints. Each sprint will have requirements listed in its respective sprint backlog and will last 2 weeks. There will be daily scrum meetings in person or over discord throughout the process.

At the end of each sprint we would have the increments working subsystem.




## 1.6 Tools

For the hardware we are going to need a server running a LAMP stack (Linux, Apache, MySQL, PhP). For the software we will need an IDE, probably visual studio and we will use javascript, php, html, css and MySQL.

## 1.7 High-Level Business Requirements

- ### Functional Requirements

Make Payment

Bill Customer

File complaint about other tenants (i.e. noise complaints)

File maintenance request

Add Property

Set Property Type

Set Rent

Contact Tenant

Add/remove tenant from property

- ### Non-functional requirements

8 week time frame

Coding skills

Secure login due to sensitive information

Store past payments and maintenance requests

Keep the expenses minimal due to the non-existent budget

## 1.8 Product backlog

**You can use the following table:**

| Priority | User Story | Tasks | Estimated effort | Sprint |
|---|---|---|---|---|
| **1** | **As a landlord, I want to be able to create an account so that I can use the service** | **Design and implement interface** | **1 H** | **1** |
| **1** | **As a tenant, I want to be able to create an account so that I can use the service** | **Design and implement interface** | **1 H** | **1** |

| | | | | |
|---|---|---|---|---|
| 2 | **As a landlord, I want to bill my tenants, So that I don't have to wait for them to pay.** | **Design and implement the interface** | **1 H** | **2** |
| | | | **1.5 H** | |
| | | | **5 H** | |
| **1** | **As a tenant, I want to log in to my account So that I can pay my rent.** | **Design and implement interface** | **1 H** | **1** |
| | | **Create tenant login** | **2 H** | |
| | | | **1 H** | |
| **1** | **As a landlord, I want to login to my account So that I can view all my properties.** | **Design and implement interface** | **1 H** | **1** |
| | | **Create landlord login** | **2 H** | |
| | | | | |
| **2** | **As a tenant, I want to file a complaint for my property** | **Design and implement interface** | **1 H** | **2** |
| | | | | |
| | | | | |
| **1** | **As a landlord, I want to be able to add/remove tenants to a property so I know who is living at a property** | **Design functionality** | **2 H** | **1** |

# 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 limitation ....etc.). 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].