|  |
| --- |
| Elaboration Report |
| **Prepared By:**  Purnima (041161250)  **Algonquin College Professor:** Asim Javaid Butt  **Date of Submission:** 2025-07-29 |

# Table of Contents

You must use proper styles for the Table of Contents to show the correct info, you must update the Table of Contents before submitting for signature.

Table of Contents 3

List of Tables and Figures (Optional) 3

Introduction 4

Project Purpose 4

Project Scope 4

Overall Description 4

User Needs 4

Assumptions 4

Dependencies 4

Requirements 4

Functional Requirements 4

Non-Functional Requirements 4

Work Breakdown Structure 4

Risks Log 5

Issue Log 6

Decision Log 6

Functional Description 6

Key Agile User Stories 7

Sequence diagram 7

Use Case Diagrams 7

APPENDIX A 8

Glossary 8

References 8

# List of Tables and Figures (Optional)

The list of tables and figures includes tables and figures (maps, graphs …). If not applicable, remove the section. If applicable, update the table below.

**No table of figures entries found.**

# Introduction

## Project Purpose

The purpose of this project is to develop DoorLoop, a fully customizable, cloud-based property management software that enables property managers, landlords, and real estate professionals to manage properties, tenants, leases, finances, maintenance requests, and communications from a unified platform.

## Project Scope

The goal of DoorLoop is to develop an all-in-one, cloud-based property management software that helps landlords and property managers streamline operations, reduce manual work, and improve tenant experience. The platform will deliver value by automating rent collection, lease tracking, maintenance requests, and financial reporting—all from a single, easy-to-use dashboard.

# Overall Description

## User Needs

* Landlords/Managers need to efficiently track rent payments, manage leases, communicate with tenants, and generate financial reports.
* Tenants need an easy, secure way to pay rent, submit maintenance requests, and access documents.
* Landlords need dashboards, analytics, and access control tools to monitor overall performance.

## Assumptions

* All users will have access to the internet.
* End users will have basic digital literacy and be able to navigate a standard web interface.
* The software does not support multi-lingual interface.
* The users have standard browser support.
* The data including tenants, payments or maintenance requests will be centralized using AWS.
* The property manager inputs initial data such as units, tenants and leases.
* There are only digital payments records which are managed by the software.

## Dependencies

* Payment gateway integration (e.g., Stripe)
* Third-party accounting tools (e.g., QuickBooks)
* Cloud storage and hosting provider (e.g., AWS, Firebase)
* Regulatory compliance

# Requirements

## Functional Requirements

The system shall:

* Let the property managers create user accounts for tenants and give them access to the application
* Let the property managers to add or update their properties (setting availability, description, images, location, price)
* Have fully customizable dashboard for both property managers and tenants
* Provide a separate tenant portal to the tenants within the application
* Have a global search feature
* Have customizable rental applications
* Have integration with Trans Union
* Have customizable Chart of Accounts for reports
* Have single menu to do everything
* Provide shortcut options
* Let the property manager access default reports like profit or loss, transactions, ledgers, tenants
* Have a feature of rent and payment related notifications
* Let the property manager enable the automated emails for late fee charged to tenants
* Let the property manager to decide the visibility of reports to the tenants
* Let the property manager to invite the existing tenants to the tenant portal
* Let the property manager to send automated rent reminders to the tenants

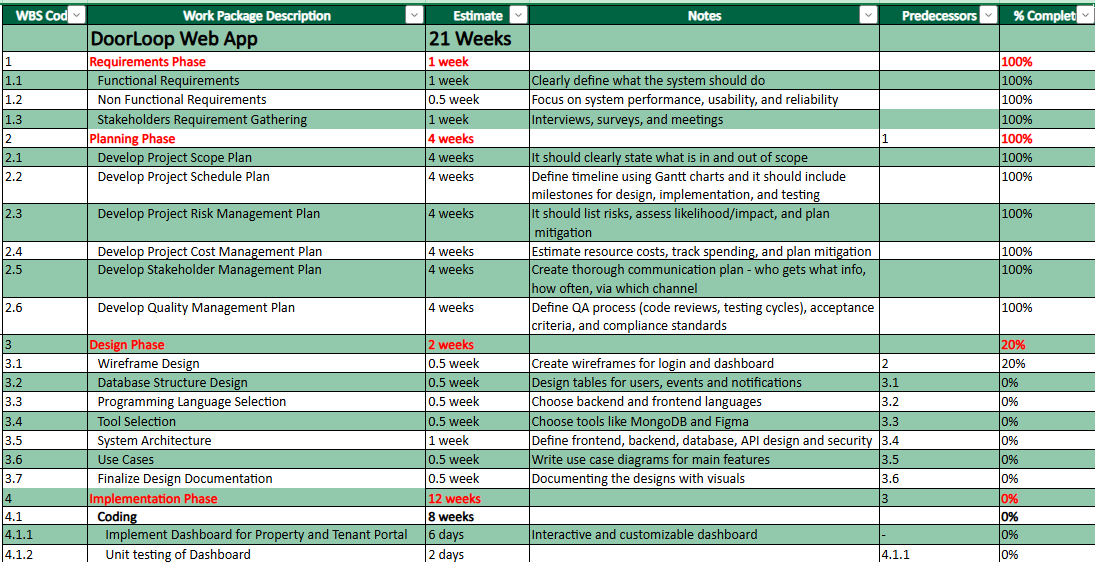
## Non-Functional Requirements

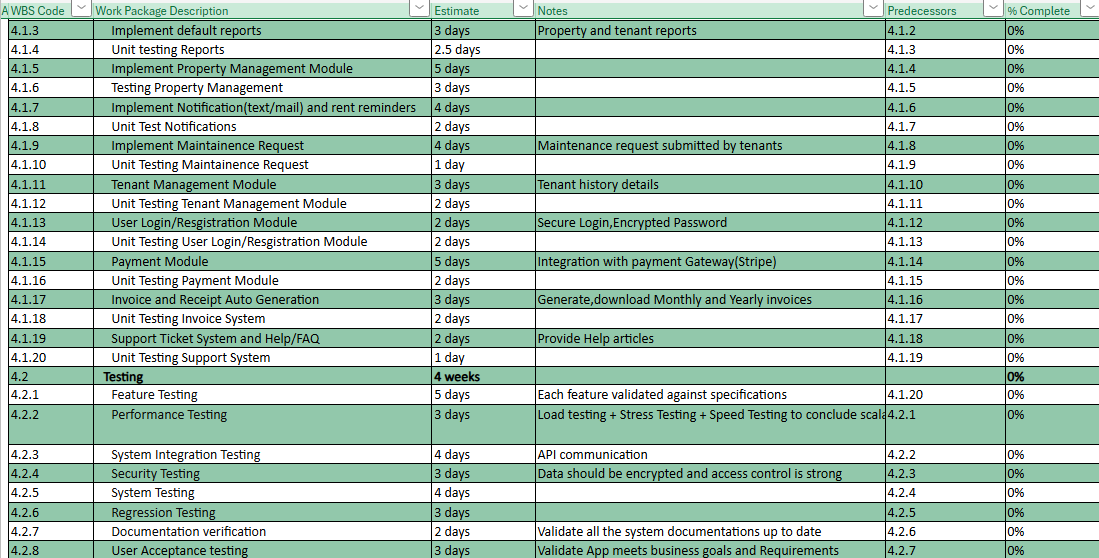
The system shall:

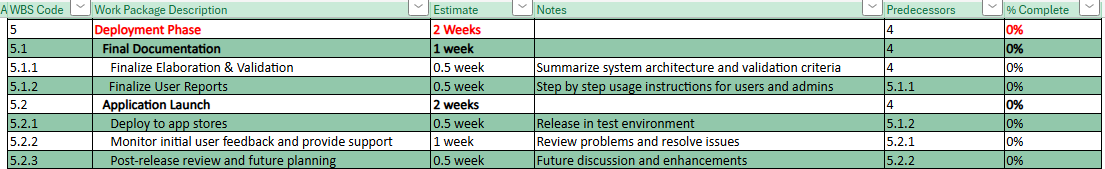
* Comply with local data protection laws
* Load within 3 seconds
* Securely store sensitive data
* Have user-friendly interface

# Work Breakdown Structure

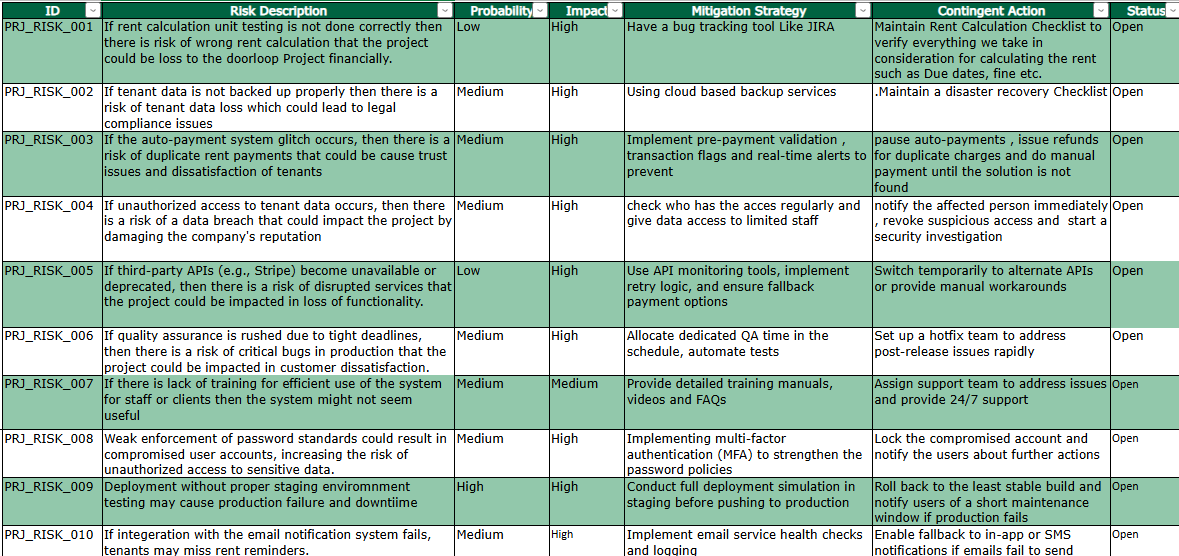
Summarized WBS which lists out the high-level plan to complete the project.

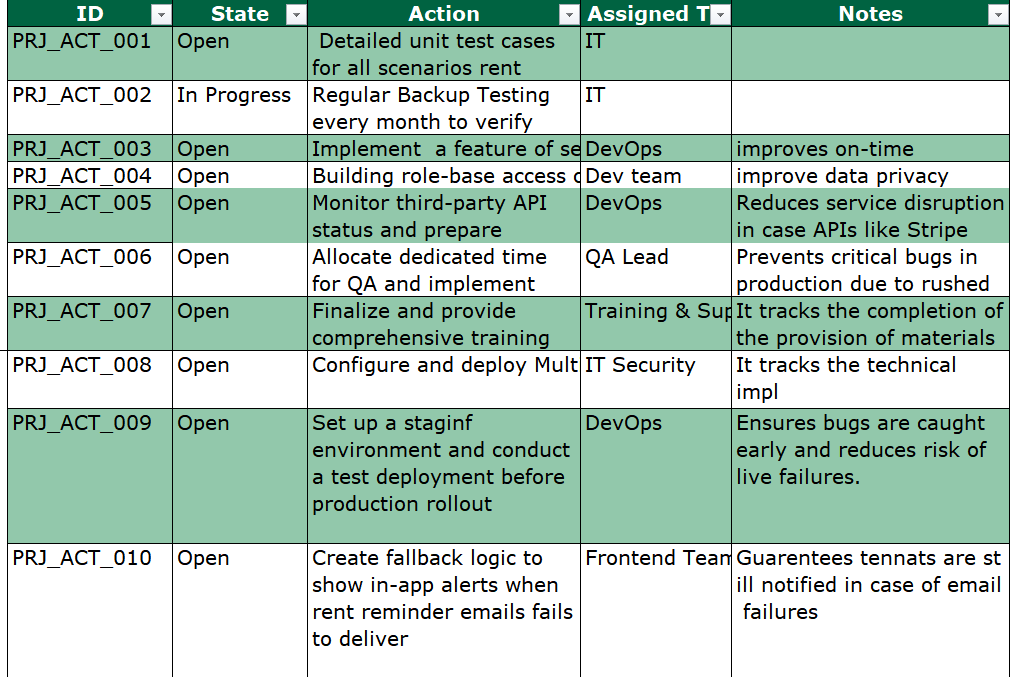






# Risks Log





# Issue Log

List of project issues, past and present. All issues are worded as if they are happening in the present, this is seen as a snapshot of the project, not all issues will yet be resolved.

A screenshot of a computer

AI-generated content may be incorrect.

# Decision Log

Lists out decisions that are reasonable to assume have been made throughout the project. Acts as a record in case anyone needs to reference back to a decision made.

A screenshot of a computer

AI-generated content may be incorrect.

# Functional Description

This section includes program design models.

## Key Agile User Stories

A table with text on it

AI-generated content may be incorrect.

A screenshot of a document

AI-generated content may be incorrect.

## Sequence diagram

One diagram required pick the most complicated timing/integration example to illustrate.

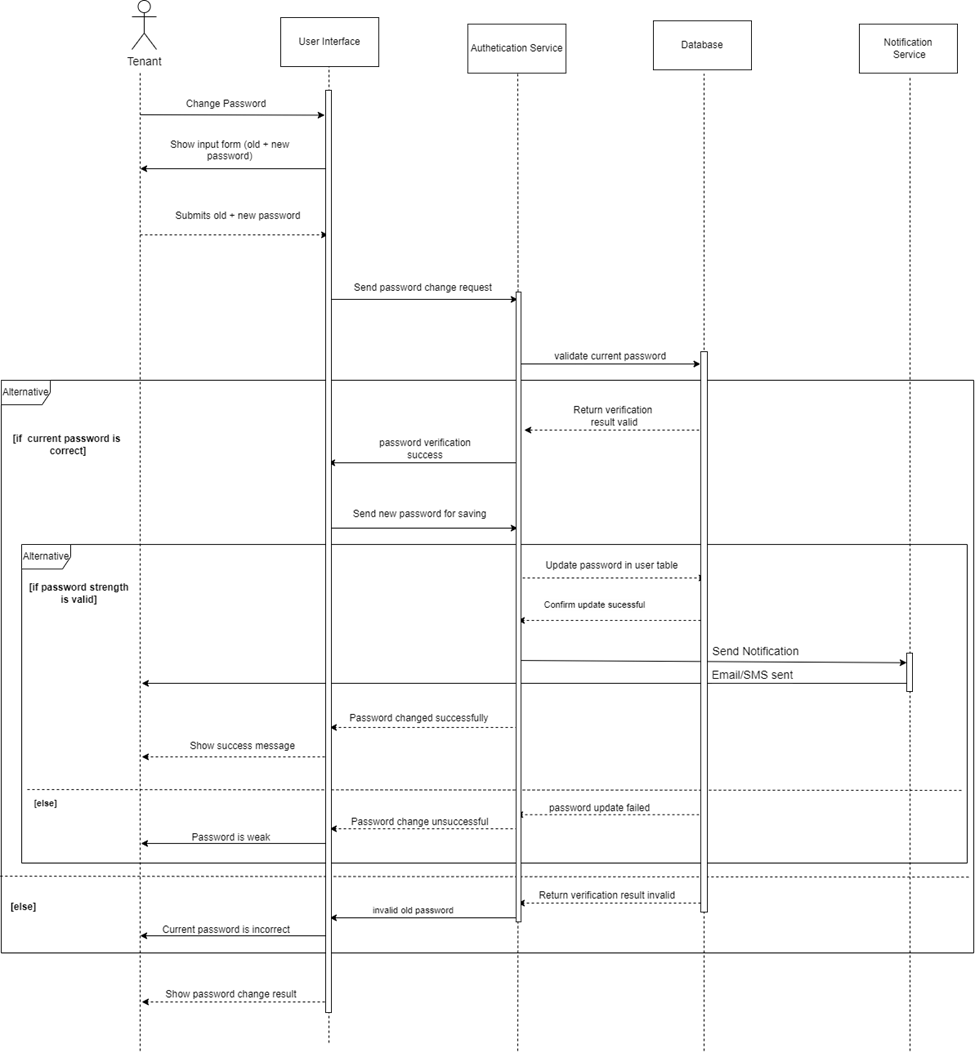
A diagram of a company structure

AI-generated content may be incorrect.

A diagram of a project

AI-generated content may be incorrect.

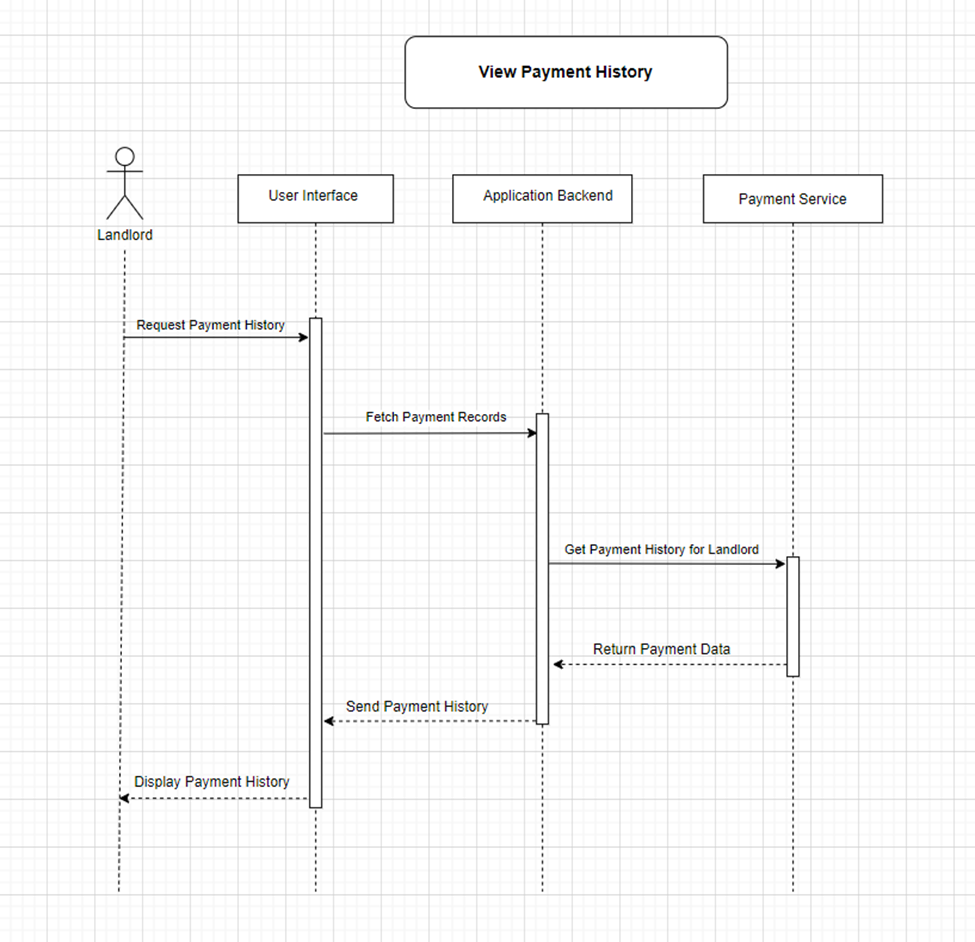
## 3.Change Password (Use-Case 3)



A diagram of a set rent payment reminder

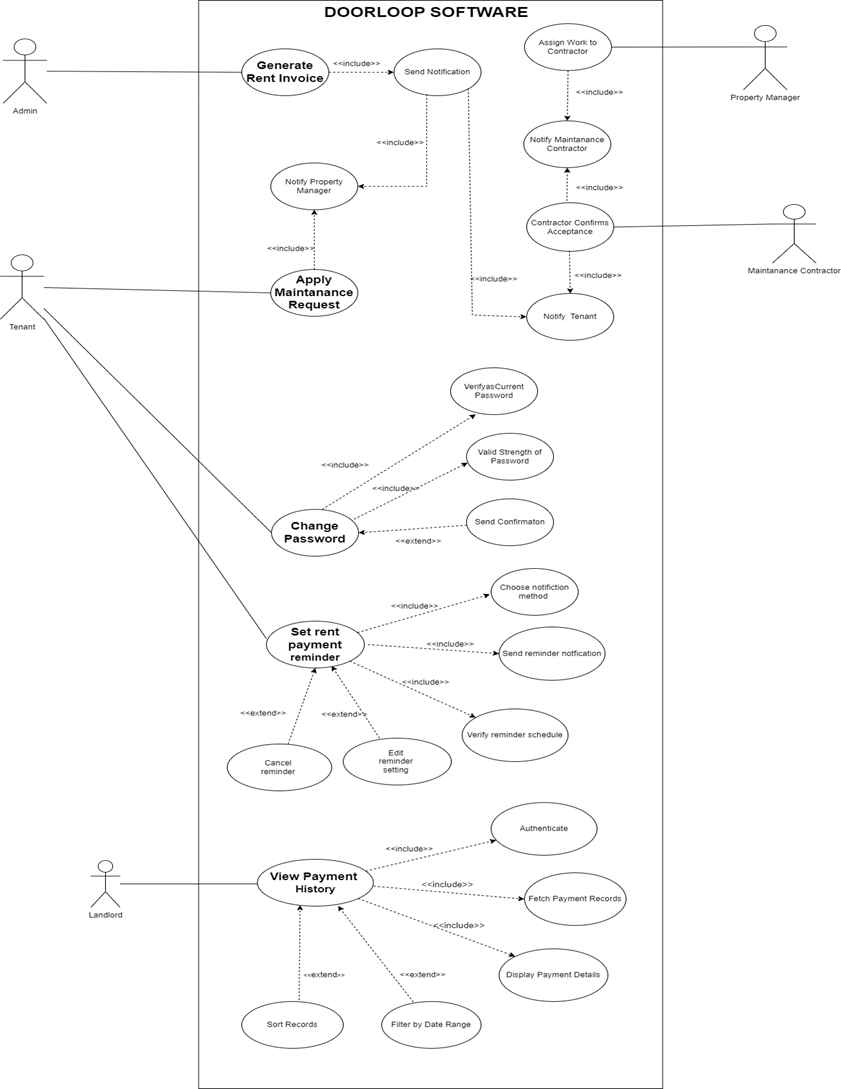
AI-generated content may be incorrect.

5. View Payment History (Use-Case 5)



## Use Case Diagrams

Three Use Case diagrams for the core functionality of the demo software.



# APPENDIX A

## Glossary

## References