

**Design Verification Plan**

**< JTT >**

|  |  |
| --- | --- |
| Name | Jacob Mexiner |
| Name | Erik Steffens |
| Name | Ken Martone |
| Name | Thaddeus Wanat |

Table of Contents

[1. Introduction 3](#_Toc401582900)

[2. Purpose and Scope 3](#_Toc401582901)

[3. References 3](#_Toc401582902)

[4. Definitions, Acronyms, Abbreviations 3](#_Toc401582903)

[5. Entrance Criteria 3](#_Toc401582904)

[6. Test Cases and Methods 3](#_Toc401582905)

[7. Resources, Equipment and Environment 4](#_Toc401582906)

# Introduction

This verification plan is to cover the purpose, scope and testing procedures that are going to be used to verify the correctness of our implementation of our property management software, ProManage. ProManage is a web based software application for the ease of managing properties and with future development of new modules with additional functionality.

# Purpose and Scope

**Purpose-**

This document describes procedures concerning the testing of the delivered products (product documents and software) of the Capstone project, ProManage, for compliance with the requirements. The requirements that the software has to be verified against can be found in the products User Requirement Documents (UR), the System Requirement Documents (SR), and Architectural Design Documents (AD). The modules to be verified and validated are defined in the AD phase. The goal of verifying and validating is to check whether the software product to be delivered conforms to the requirements of the client (the students) and to ensure a minimal number of errors in the software. This project document is written for managers and developers of this Capstone project

**Scope-**

ProManage is an application designed and developed by the “JTT” group for the Department of Engineering and Computer Science at the University of Wisconsin-Milwaukee. The purpose of the application is to assist and organize a robust and effective rental property management application. The build includes a unique architectural design including a modular design for ease of additional features as well as new modular rollouts of new platforms. The scope of this project is limited because of its duration is limited to a single semester. It describes how the verification will be done based on which requirements input it is given, who completes the verification activities, and how the traceability will be established amongst the team. Please note that as the project content could change over time, this document could be adjusted to reflect these changes.

# References

The following table lists other documents referenced by this document.

| Document |
| --- |
| User Requirement reference- UR |
| System Requirement reference- SR |
| Architectural Design reference- AD |
|  |

# Definitions, Acronyms, Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
|  |  |
|  |  |
|  |  |

# Entrance Criteria

Our entrance criteria will be the completion of our final sprint 3 deliverables for this class. Our auditing and code reviews will indicate that the deliverables promised have been completed and are ready for official testing.

The following items must be completed prior to the start of verification testing:

* All test hardware platforms must have been successfully installed, configured, and functioning properly.
* All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
* All the standard software tools including the testing tools must have been successfully installed and functioning properly.
* All personnel involved in the testing effort must be trained in the tools to be used in the testing process.
* Proper test data is available.
* The test environment such as, lab, hardware, software, and system administration support should be ready.

# Test Cases and Methods

**Code Reviews-**

*Internal Reviews*- The simple internal review is used for documents which will not reach the client directly

*External Reviews*-The external review is used to see if the document matches client expectations. The external review can only be done after the document has been accepted in an internal review.

**Audits-**

Audits are reviews that assess compliance with software requirements, specifications, baselines, standards, procedures, instructions, codes and licensing requirements. Physical audits check that all items identified as being part of the configuration are present in the product baseline. A functional audit checks that unit tests, integration tests and system tests have been carried out and records their success or failure

**Testing-**

The team will create and execute System Requirements verification test cases constructed from the System Requirements.

The results of the verification will be documented as follows:

* Environment used to execute the test: Version of SW, labeled SW, x-Cut, build number, document revision, etc. that exactly specifies the test unit
* ***Who*** executed the test
* ***When*** was the test executed
* ***Result***, why the verification failed or was not executed.  
  The observed result will be documented as a rational, why the test was passed, failed, or not executed.
  + Passed: A Passed result indicates that all preconditions were satisfied and the test case passed
  + Failed: A Failed result indicates that the test case failed or the preconditions of the test case could not be satisfied
  + Not Executed: A Not Executed result indicates that test case was intentionally not executed. The tester is required to explain why the test case was intentionally not executed in the Comments section of the test case.

# Resources, Equipment and Environment

* Python 2.7
* Django 1.8
* PostgreSQL database
* Django Test suite
* Github for version control
* Slack Communication system
* Server hosted internally