Class Project: [TITLE]

[NAME(S)]

CIS 350: [SEMESTER]

Table of Contents

Project Information 4

Features 4

Screenshots 4

Project Plan 5

Requirements & Definition 5

Development 5

Verification 5

Maintenance 5

Umbrella Activities 5

Responsibilities 5

Requirements & Specification 6

Use-Cases 6

Natural Language Requirements 6

Natural Language Requirements 6

Design 7

Development 8

Code Standards 8

Static Analysis 8

Code Documentation 8

Configuration Management 8

Verification 9

Integration Tests 9

Unit Tests 9

Code Coverage 9

Requirements Coverage 9

Postmortem 10

Earned Value 10

Variances 10

Lessons Learned 10

References 11

# Project Information

[Project Description]

## Features

[Project Features Included]

## Screenshots

[Screenshots of Project]

# Project Plan

[Overview of Project Plan, including any customizations to the software process model]

## Requirements & Definition

[Plan for requirements and definition detailed here]

## Development

[Plan for development / implementation detailed here]

## Verification

[Plan for verification detailed here]

## Maintenance

[Plan for maintenance (e.g., problems found by prof or during updates for next release) detailed here]

## Umbrella Activities

[Plan for project management and status tracking / meetings detailed here]

## Responsibilities

[Describe the responsibilities of each member, how you plan to break down the work]

# Requirements & Specification

[Description of methods used (e.g., use- case diagrams, user stories, use case-descriptions)]

## Use-Cases

[Include use-case diagrams AND use-case descriptions here]

## Natural Language Requirements

[Include requirements here]

## Natural Language Requirements

[Include traceability (matrix) from requirements to use cases]

# Design

[Description of methods used (e.g., class diagrams)]

[Design specifications]

# Development

[Description of methods used (e.g., Checkstyle, FindBugs, Javadoc) AND any additional libraries that you used]

## Code Standards

[Include report on code standards and justification for any variances]

## Static Analysis

[Include report on static analysis and justification for any variances]

## Code Documentation

[Include link to javadocs (likely included as separate file) and justification for any non-documented areas]

## Configuration Management

[Include link to Git repository and Git log]

[Explain / describe method for tracking releases]

# Verification

[Description of methods used (e.g., integration & systems and/or unit testing)]

## Integration Tests

[Include manual and integration test procedures]

## Unit Tests

[Include references to unit tests in code]

## Code Coverage

[Include code coverage reports, must include: coverage of automated tests, coverage of manual tests, and combined coverage]

## Requirements Coverage

[Include traceability (matrix) from requirements to test procedures]

# Postmortem

[Include a reflection on how well the project has gone thus far]

## Earned Value

[Include the earned value calculations for your current status and any explanation of over/under runs]

## Variances

[Include any additional variance (time, coverage, functionality, …) explanations necessary]

## Lessons Learned

[Include lessons learned]

# References

[Include references here]