System Architecture Document

Planr, an Agile Project Planning Application

Version 2.0

Submitted in partial fulfillment of the requirements of the degree of Master of Software Engineering

Prepared by

Michael Blakeman

CIS 895 – MSE Project

Kansas State University

Table of Contents

[1. Introduction 3](#_Toc86837070)

[2. Architecture of the System 3](#_Toc86837071)

[2.1 Client Application UML Diagrams 3](#_Toc86837072)

[2.2.1 System UML Diagram 3](#_Toc86837073)

[2.2.2 PlanningCoordinator UML Class Diagram 3](#_Toc86837074)

[3. Class and Method Descriptions 4](#_Toc86837075)

[3.1 Project 4](#_Toc86837076)

[3.2 Engineer 4](#_Toc86837077)

[3.3 PlanrFeature 4](#_Toc86837078)

[3.3.1 UnplannedFeature 4](#_Toc86837079)

[3.3.2 PlannedFeature 4](#_Toc86837080)

[3.4 Sprint 5](#_Toc86837081)

[3.5 WorkBlock 5](#_Toc86837082)

[3.6 Platform 5](#_Toc86837083)

[3.7 Roadmap 5](#_Toc86837084)

[3.8 PlanningCoordinator 5](#_Toc86837085)

# Introduction

This is the architecture design document for the Planr application. This document provides a reference to the design of the application with class and sequence diagrams. Each class will have a brief description about the function of the class and its methods.

# Architecture of the System

## 2.1 Client Application UML Diagrams

### 2.2.1 System UML Diagram

Diagram

Description automatically generated

### 2.2.2 PlanningCoordinator UML Class Diagram

A picture containing graphical user interface

Description automatically generated

# Class and Method Descriptions

## 3.1 Project

This struct is a representation of an unplanned body of work containing Engineers and UnplanedFeatures. It is the top-level object that gets passed into the PlanningCoordinator to plan and used to generate the Roadmap. This has methods to add and remove UnplannedFeatures and Engineers.

## 3.2 Engineer

This struct is a representation of an Engineer for planning purposes. The constructor requires a first name, last name, a collection of platforms the Engineer is proficient in, and a collection of dates that the Engineer is unavailable to work. There is a validate method on this struct that can be used to validate the input from the user. This will be helpful for unit testing and error handling.

## 3.3 PlanrFeature

This is an interface (or protocol in Swift) that both the UnplannedFeature and PlannedFeature classes implement. This includes the name of the feature, a summary of the feature, the platforms that apply, an effort estimate (per platform), a priority value from 0-1000, a Boolean value for if concurrency is allowed for the feature, a color to display on the UI, and a feature id GUID to identify the feature.

## 3.3.1 UnplannedFeature

This class conforms to the PlanrFeature protocol and represents an unplanned feature. A collection of these are passed into the Project contstructor to be used for planning. This has a public method to update the name of the feature and another public method to update the priority of the feature. As with other classes / structs in the project there is a public validate method to aide in error handling and unit testing.

## 3.3.2 PlannedFeature

This class conforms to the PlanrFeature protocol and represents a planned feature. A collection of these are constructed from the plan() method in the PlanningCoordinator and are assigned to a 1:1 ratio to WorkBlocks by being passed into the WorkBlock’s constructor. A planned feature has a collection of associated WorkBlocks.

## 3.4 Sprint

This structure represents an Agile Sprint. The constructor takes in a collection of WorkBlocks, the points remaining in the sprint, and a date range of the start and end dates of the sprint. There is a public method to add a WorkBlock to the sprint and a public method to output the information about the sprint in the console for debugging purposes.

## 3.5 WorkBlock

This struct is a representation of the block of work inside a sprint. It requires a name, a single platform, and a point value to be initialized. It has two constructors one that takes in a PlannedFeature and another that takes in the individual parameters. There is a public method to assing a sprintId to the work block for association purposes.

## 3.6 Platform

This enum is used to indicate what platform to use for the given associated struct or class. This is used throughout the application and is a property on a number of structs.

## 3.7 Roadmap

This struct represents a planned timeline of sprints. The constructor takes in a collection of Sprint instances and has a public method for printing the roadmap information to console.

## 3.8 PlanningCoordinator

This struct is used to plan the Roadmap. The constructor takes in the Project, the start date for planning, an average velocity of the team, the length of sprints in weeks, and an estimate padding multiplier in the form of a Double. The plan() public method is used to plan the project and will output a Roadmap object that will be used to display on the UI.