MovShare

Project Overview

This project aims to build an app that helps people manage and keep track of different kinds of media owned by themselves or others.

In order to reduce unneeded purchases and waste time this app will allow a group of users to store titles and details relating to their media collection. Each user will them be able to search for titles in each others libraries in order to share this media amongst themselves. This would also allow someone to see what another doesn't have in order to purchase a gift or bring something over when visiting.

The app could gain multiple additions, such as a wishlist for users, and the ability to scan a barcode from an App, or manually enter in a UPC in order to search for products.

Team Organization

The team is setup of a project manager, scrum master, and team members. Each member of the group will act as a team member and switch between acting project manager or scrum master.

For phase 1 the project manager will be Mitch, and the scrum master will be Garett. This is planned to switch about every phase of the project.

Software Development Process

The development will be broken up into five phases. Each phase will be a little like a Sprint in an Agile method and a little like an iteration in a Spiral process. Specifically, each phase will be like a Sprint, in that work to be done will be organized into small tasks, placed into a "backlog", and prioritized.   Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team will use a Scrum Board to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, and those that are done.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

Phase	Iteration
1.	Phase 1 - Requirements Capture
2.	Phase 2 - Analysis

3	Phase 3 - Architectural, UI, and DB Design
4	Phase 4 - Detailed Design, Implementation, and Unit Testing
5	Phase 5 - More Implementation and Testing

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

Communication policies, procedures, and tools

The team will communicate mainly via Slack, a collaboration hub application, and in person meetings.

Quick, 10 minutes or less, meetings will occur at least three times a week usually after class. Longer meetings, ranging from 30-90 minutes, will be scheduled at least once a week Wednesday-Friday in the evening or on Saturday.

Project progress will be communicated through a combination of Slack and the Github scrum board.

Risk Analysis

Potential risks involve time restrictions, team availability, health issues, development skills, and tool reliability.

Time and team availability will be the most likely to happen, and therefore the most expected. As we are able to properly expect these limitations and risks we will be able to plan around them and have minimal impact towards the final product.

Health issues could result in team members being unable to do their tasks for a period of time. These risks have a low likelihood of presenting themselves, however they could produce a larger impact to the project. By planning the project in a way where each team member has a little extra availability we will be able to handle unexpected absences due to health or other issues the would cause a team member to be absent.

Development skills and tool reliability are areas that have a high likelihood to appear and can have a major impact to the project. However, these are both risks that are easily avoided by assigning tasks to the proper team member and ensuring tools are properly setup and requirements defined before development starts.

Configuration Management

See the README.md in the Git repository.