Rent-A-Movie

**Project Overview**

This project aims to build an app that helps people minimize duplicate purchases of movies by allowing users to post and rent out movies to other users.

Have you ever purchased a movie online then discovered that a family member or close friend just brought the same thing? To reduce redundant purchases, we will make a web-based app that allows you, family members, and close friends with whom you often watch movies with to be able to browse each other’s lists of titles. The app would allow you to add, update, and delete the titles in your own list, allow others to see your list, invite them to share their lists, and seamlessly browse or search all lists to which you have access to. The app would not allow you to share the actual media but will allow you to request to meet with them to rent their movies. In order to notify the users of rental requests and meet-up details, the app will maintain a email address associated with the user profile. To maintain confidentiality and security, the app would authenticate users and allow them to manage access rights they have granted to others.

**Team Organization**

The team consists of four members: George Done, Zach Hawkes, Tanner Hooper and Nick Jugganaikloo.

The team will be organized into one scrum master (also a product owner) and other individual team members. Both scrum master and team members work on various aspects of the project. The role of the scrum master will be rotated to a different member at the beginning of a new phase. The work will be divided among team members by creating issues in GitHub and then a team member will assign himself to an issue that he wants to work on.

**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**

Google hangouts will be the main way of communication between the team members. Google drive will also be used to share ideas and documents that would make sense to be shared outside of the git repositories (UML diagrams, etc.). Email notification from GitHub may also be used to remind each other to review the pull requests.

**Risk Analysis**

Project Risks

* Team member(s) not having skills/knowledge to complete different tasks.
  + Train members and provide assistance when needed.
* Team is able to meet deliverable deadlines.
  + Frequently check-in with each members progress and provide suggestions/assistance when needed.
* Project code is entirely erased.
  + Frequently pull from master repo to have local backups.
* Team member(s) falls ill/succumbs to death.
  + Ask Project manager for inspired guidance.

**Configuration Management**

See the README.md in the Git repository.