CS39440: Project Outline

VIDEO GAME BACKLOG APPLICATION

Department of Computer Science,
Aberystwyth University

Last updated: **12th February, 2024**

v1.0 - RELEASE

Produced by:

Kal Sandbrook
kas143@aber.ac.uk
on *Computer Science - G400 BSc*

Supervised by:

Dr. Edore Akpokodje
eta@aber.ac.uk
Lecturer in Computer Science

Contents

1 - Project description

This project aims to develop an application which helps users manage their video game collections in an effective manner. The key goal is provide users with a platform to curate and view their games, offering features such as: backlog management, progress tracking and powerful sorting and filtering options.

Users will have the ability to mark a game with various states (such as “abanadoned”, “completed” or “playing”), allowing for easy organization and awareness of a users gaming endeavours. Additionally, a user will be able to sort and filter games by information like release date, genre and platform. A tagging system will also be implemented to further aid categorization based on user preferences.

In the initial stages of development, users will manually input their game data into the application. However, later stages will see an API being integrated to automatically fetch game information, providing a more seamless experience.[1]

As the project progresses, additional features that will be explored include: a recommendation system that suggests games based on the user’s backlog and completed games; as well as a social sharing feature to allow users to share information about their games with others.

Regarding the development platform, whilst a final decision has not been made, the initial intention is to develop a native desktop application. Further research will be conducted to determine the optimal platform, with the likely candidates being Python (PyQt [1]) or Java (QtJambi [2]).

The goal of the project is to create a user-friendly and feature-rich application that caters to the needs of gaming enthusiasts, providing a comprehensive tool to manage and enhance the games that they are playing.

A variation of the Kanban agile methodology will be used to develop this project, with tasks being visualised on a kanban board.

2 - Proposed tasks

The major tasks that will need to be undertaken are as follows:

- *Research and Evaluation of Development Platforms*
  - Investigation into the suitability of different platforms for development of the application. Factors to be taken into account include the features, capabilities and support of each language, along with suitability for an object-oriented design.
  - This will involve making small test programs on the platforms in question, to get a feel for the suitability of a language.
- *Set up of local environment and version control*
  - An environment will need to be configured for development of the project. The specifics of this environment will depend on the chosen platform.
  - The project will use a Git repository for its version control. This will either use the University provided GitLab [3] instance, or the external service GitHub. [4]

[1] The application will allow the user to choose which API to use, and will not be limited to a single API.

- **Development**
  1. *Initial Development and Design*
     - Early development will involve building the essential functions of the application. A rough list of initial duties will involve: designing the database schema, implementing basic CRUD[2] functionality, designing the User Interface, the ability to manage the backlog and filter through it. At this stage, game data will be provided by the user.
  2. *API Integration and Categorisation*
     - This stage of development will involve making use of external APIs to fetch game information and adding a tagging system to allow users to further categorise their games.
  3. *Additional Features and Testing*
     - The final stage of development will involve the research and development of the additional features outlined in Section 1, time permitting. This stage will also include a focus on testing, to ensure the application is of sufficient quality.[3]
- *Project Meetings & Kanban Board*
  - Throughout the project, weekly meetings will take place with the supervisor. The agenda for these meetings will be aided by a kanban board showing work that is yet to commence, underway or completed. This kanban board will use the GitLab Issue Board[4] feature or the GitHub Project feature[5]. This will allow for a good visualisation of "work done" over each week.
- *Demonstrations*
  - There are two demonstrations scheduled for this project. The first after two months of development, and the second at the end of development. For the first demonstration mid-project, it is planned that the initial development stage (detailed above) will be implemented.

3 - Project deliverables

The project will deliver the following items throughout and at the end of development:

- *Demonstration Notes*
  - A set of notes that describe what was shown during the first demonstration. This will be included in the appendix of the *Final Report* and a draft will be produced beforehand.
- *Working Software Application*
  - A fully functional video game management application capable of managing user game collections, including backlog management, progress tracking, sorting, filtering, and tagging features. This will be provided in the form of an executable file or installer for Windows and potentially Linux.

[2] Creating, Reading, Updating and Deleting.

[3] Testing will also taking place during development, via unit testing and potentially a Continuous Integration system

[4] Details can be seen at: https://docs.gitlab.com/ee/user/project/issue_board.html

[5] Details can be seen at: https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects

- **Final Report**
  - This report will discuss the work and acknowledge any 3rd party tools used in the project. It will contain various appendices of documentation produced during development, including:
    - Platform Investigation Report
    - User Interface Mockups
    - System Design Documentation
      - There will be documentation included that detail the Database Schema, External API Integration, Recommendation System and Sharing Feature.
    - There will be a testing and quality assurance document included, showing the results of the tests carried out during development.
- *Final Demonstration*
  - This will not produce any documentation, but it is a deliverable that should be considered in planning.

Bibliography

[1] "PyQt". Accessed: Feb. 07, 2024. [Online]. Available: https://wiki.python.org/moin/PyQt

[2] "QtJambi Repository". Accessed: Feb. 07, 2024. [Online]. Available: https://qtjambi.io/

[3] Prifysgol Aberystwyth University, "Aberystwyth University - GitLab". [Online]. Available: https://gitlab.aber.ac.uk/

Similar to GitHub, but hosted by the University. However, it has a more limited set of features for tracking development.

[4] GitHub Incorporated, "GitHub". [Online]. Available: https://github.com/

An external service that offers the ability to host private git repositories. It also allows for Kanban-like functionality and issue tracking to aid development.

[5] "IGDB API Documentation". Accessed: Feb. 07, 2024. [Online]. Available: https://api-docs.igdb.com/

One of the APIs that may be supported by the application. It does not include as many games as the Steam WebAPI, but may be more in depth.

[6] "Steam WebAPI Documentation". Accessed: Feb. 07, 2024. [Online]. Available: https://wiki.teamfortress.com/wiki/WebAPI

One of the APIs the application may support. It includes many games, but available information may be limited.

[7] "KDE Human Interface Guidelines". Accessed: Feb. 07, 2024. [Online]. Available: https://develop.kde.org/hig/

A set of principles that will be taken into account when designing the User Interface.