Complacara

Conundrum

Project Report

David Berthod-Perez, Ethan Thompson, Ethan Peeler

Capstone Project Fall, 2023

The University of West Florida

12/1/2023

CIS 4595

Dr. Owsnicki-Klewe, Bernd

Project description

Our project establishes a simple Capstone goal of transforming the "Big-Five" personality assessment into an engaging and enjoyable game format. By gamifying personality, we offer a more stimulating and less biased experience when compared to the usual questionnaire, producing a broader and friendlier experience for the audience. Ultimately this project delivers a comprehensive assessment in the form of an easy and accessible choice-based web adventure game, enhancing both accuracy and user engagement in determining personality. The program itself was created mostly using ReactJS, Node/ExpressJS, Replit, and Firebase.

Table of Contents

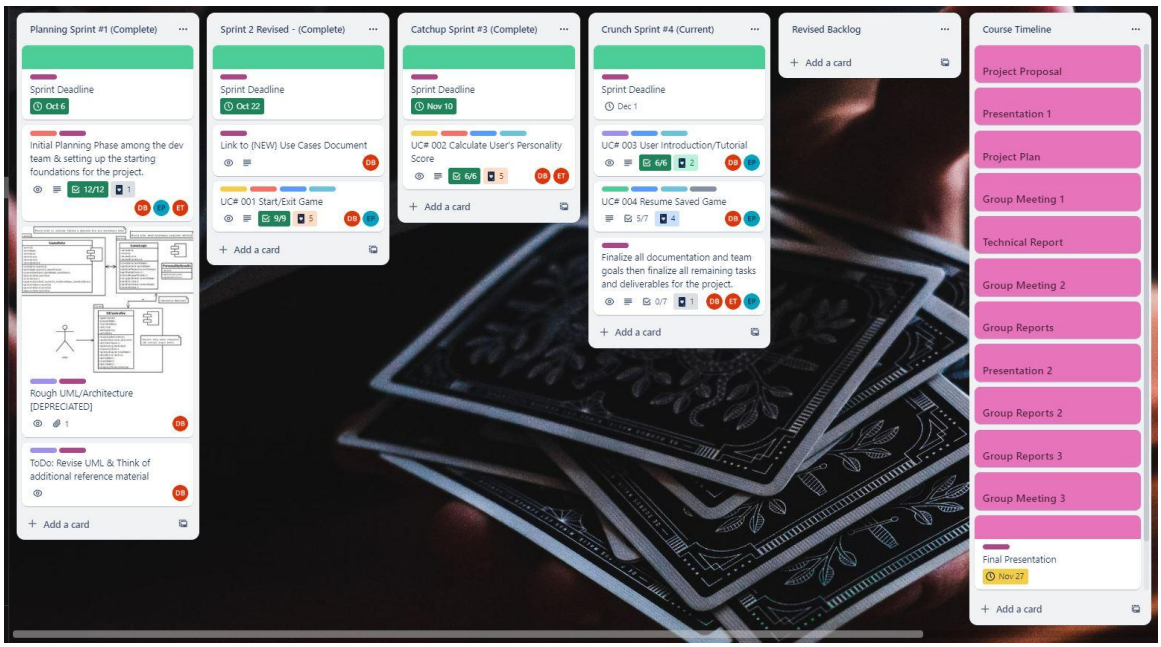
Table of Contents

List of Figures

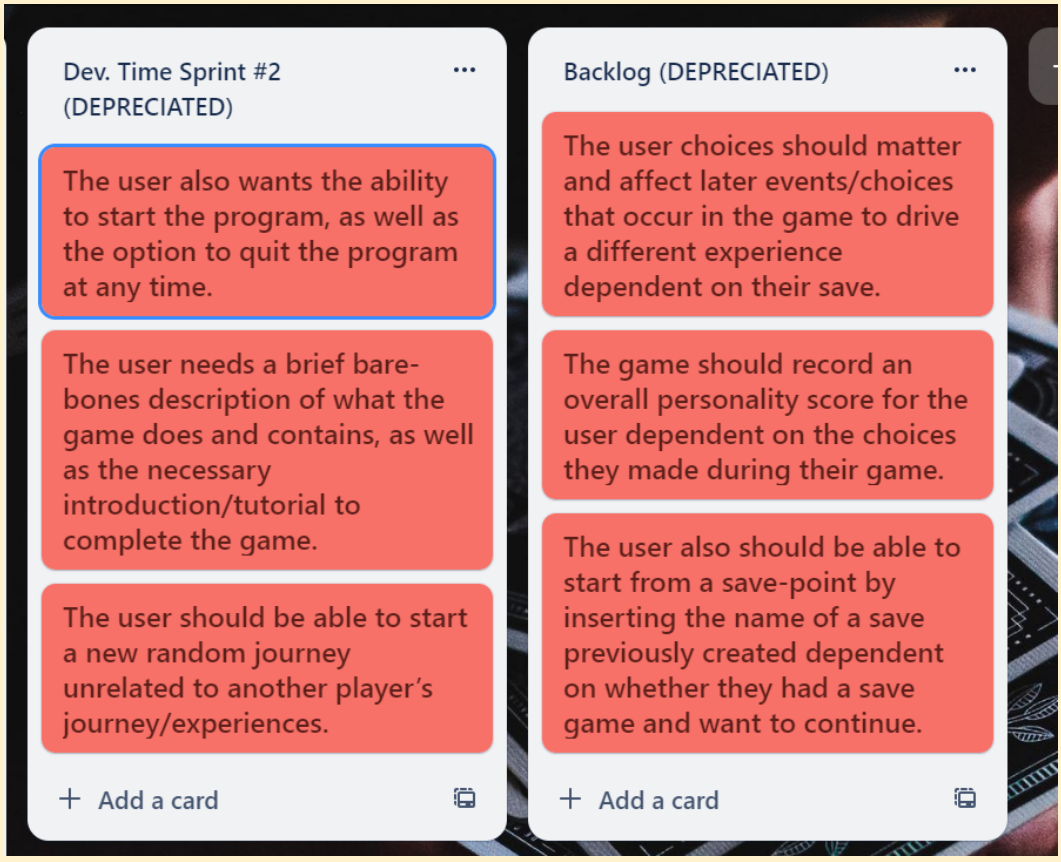
List of Tables

1 Final timeline and comparison with the initial timeline

Our initial timeline was a bit ambitious. We had fewer sprints that were more comprehensive and involved. We also initially had some downtime in between sprints. This would have been a fantastic schedule if we had been able to keep it, as it would have given us time to be a bit more imaginative with our project. However, our time was limited by the scope of the class and our goal was to release a competent, working project. The images below show a piece of our original timeline as well as our actualized timeline.



Current Timeline



Depreciated Timeline Parts

2 Project results compared with expectations

Implemented use cases according to Technical Documents and results:

2.1 UC# 001: Start/Exit Game

2.1.1 Expected Use Case

Summary:

As a user, I want the option to start the game allowing me to begin a journey or interact with the menus and exit it at any time to provide flexibility and convenience in engaging with the game.

Use Case:

Element	Description
Priority	MUST
Actors	User, Game Server
Goal	The user can start the game or exit it at will.
Preconditions	The game is running, and the user has access to the game interface and application.
Postconditions	The game is either started, allowing the user to play, or it is exited, and the game session terminates.
Use Case	The user initiates the game, and the game server handles the startup process and available adventure options. At any point during the game, the user can choose to exit the game, which will save their progress if needed.

2.1.2 Use Case Outcome

1. A user enters the game and a starting score is set based on character creation.
2. From there, the user makes choices that affect all OCEAN Big 5 scores differently (Note: Some journeys may vary more than others based on decisions).
3. When the user reaches an endgame condition, the overall average of scoring all categories is calculated and then scaled to each category to determine a personality summary for that field.

2.2 UC# 002: Calculate User's Personality Score

2.2.1 Expected Use Case

Summary:

As a user, I want the game to calculate and record my overall personality score based on the decisions and choices I make throughout the game to provide a unique and personalized outcome.

Use Case:

Element	Description
Priority	MUST
Actors	User, Game Server
Goal	The game calculates and records the user's personality score based on their in-game decisions.
Preconditions	The user has made choices and decisions within the game.
Postconditions	The user's overall personality score is calculated and recorded in the game database.
Use Case	The user's in-game choices are processed by the game server, and the resulting personality score is stored for future reference.

2.2.2 Use Case Outcome

1. A user enters the game.
2. Whenever the user makes a choice the effects on the score are continuously stored and processed for future calculation.
3. When the user reaches the end of the game or the limit, they receive their overall final score noting all actions taken place.

2.3 UC# 003: User Introduction/Tutorial

2.3.1 Expected Use Case

Summary:

As a user, I want a clear and engaging introduction to the game that will help me understand its purpose and help me progress through the game. As an option, I want to be able to skip this tutorial if I am already familiar with the game.

Use Case:

Element	Description
Priority	SHOULD
Actors	User, Game Server
Goal	The user receives a clear and engaging introduction and tutorial to understand the game's purpose and mechanics.
Preconditions	The user has launched the game.
Postconditions	The user has a basic understanding of the game's concept and mechanics.
Use Case	The game server provides the user with a tutorial or introduction explaining the game's narrative, choices, and how to progress in the game. Optionally: This tutorial can be skippable.

2.3.2 Use Case Outcome

1. A user enters the game.
2. When the game starts the first scene should be a tutorial with the options to follow along or skip for the player.
3. The player can proceed through the tutorial and game without issues.

2.4 UC# 004: Resume Saved Game

2.4.1 Expected Use Case

Summary:

As a user, I want to be able to resume my progress in the text-based adventure by entering the name of a previously saved game, allowing continuity in my journey of self-discovery and personality analysis.

Use Case:

Element	Description
Priority	COULD
Actors	User, Game Server
Goal	The user can resume their progress in the text-based adventure by entering the name of a previously saved game.
Preconditions	The user has a previously saved game, and the game is running.
Postconditions	The user continues the game from the point where they left off.
Use Case	The user provides the name of the saved game, and the game server loads the corresponding game state, allowing the user to continue their adventure.

2.4.2 Use Case Outcome

1. A user enters the game.
2. From there the user can proceed through the tutorial and start their adventure.
3. The user can use the menu to load a previous game (if in the current test session, currently missing logic to upload/pull saves to/from the database).
4. If they have no save to load, then they can proceed through a point in the game they wish to save through the menu and load it at any time, keeping their scene position and score from that moment.

3 Software evaluation

3.1 Functionality

The main test of functionality was undergoing penetration testing. We wanted to see what users could do to attempt to break the software and produce unexpected output.

3.1.1 Testing Strategy

We used several different testing techniques. We utilized integration, unit, and user acceptance testing. We used integration testing to make sure that all of the pieces of our website worked together. The user acceptance testing came by displaying our project to potential users and getting their feedback on it.

3.1.2 Test plan

Our testing plan was relatively simple. We know that the code we write is essentially a conglomeration of primarily other people's code (through external libraries) and a minority of our own code. Therefore, we attempted to ensure that the external libraries we were using were regularly updated and secure. This was our main focus.

3.1.3 Test cases/data

One of our test cases was to simulate a user creating an account on the website. We put in all the information a user might need to fill in the account and see if we were able to create an account and if that information was stored in Firebase. We also tried to create an account with data that should not work.

3.1.4 Observed defects/failures

There is a bug that causes the game data not to be loaded correctly if the user has either not passed the tutorial or has not skipped it yet.

Also, we had an issue where we could indefinitely choose an option, which broke the test. However, that seems to now be fixed with our latest release.

3.2 Security

We have implemented fuzz testing, seeing if we can input invalid data into the input fields and have the application accept it. We also minimized our usage of text boxes so that way it is both easier for the end user and the possibility of unexpected input is minimized.

3.2.1 Security requirements

The web application must keep the user's password confidential, creating a zero-knowledge architecture.

3.2.2 Data protection method (Encryption and Hashing) details

Firebase took care of our authentication for us. It uses a key derivation function known as Scrypt to both hash and salt our authentication information.

3.2.3 Code verification

We not only had peer code reviews, but we also had AI review our code as well. This included evaluating the quality of the code as well as suggesting areas for improvement.

3.2.4 Security testing

We performed a fairly front-facing penetration test on our software to attempt to find both any bugs and any possibilities of possible unauthorized access or modifications to our software.

3.2.5 Summary

Our project is quite robust to attack. We attempted to reduce the number of attack vectors when possible. This was especially executed for the input forms. We wanted to have as little textual user input as possible, instead replacing text boxes with radio dropdown menus where possible. There is also client-side validation that helps with the UX.

4 Work to be done

- Touching up database usage and protections.
- Bug hunting, testing, and fixing.
- Develop longer and more verbose storylines or LLM integration.
- Implement more robust decision analysis (e.g. Karma System).
- Get contact/Involvement with a Psychology Subject Matter Expert.

5 Acknowledgements

This work was conducted under the Fall, 2023 Capstone assignment administered by UWF's Hall Marcus College of Science and Engineering. The members of NoCap are also particularly grateful to Dr. Owsnicki-Klewe for his insight into the nature of the Capstone roadmap.

6 References

[1] N/A

Appendix A: N/A

N/A