Department of Computer Science & Engineering, University of Nevada, Reno

BETA Universe Systems Initiative Table Application

Team 7: Taking Initiative

Jacob Gayban, Mark Graham, Andy Alarcon, Jacob Tucker, Griffin Wagenknecht

Dr. David Feil-Seifer, Vinh Le, Dr. Devrin Lee

John Molt

February 4, 2022

Table of Contents

Abstract

In tabletop role-playing games, players can be burdened by extraneous calculations and player order, which disrupt the flow and narrative of the game. The BETA Universe Systems Initiative Table Application is a mobile-optimized companion web application that aims to digitize and streamline gameplay components from the tabletop role-playing game "BETA Universe Systems." Players can register for accounts, manage player characters, and participate in game sessions that feature real-time communication via a chat log and combat loop. Gamemasters can also register for accounts, manage the non-player characters and manage game sessions along with running them. This document aims to outline the project's concept and the project management throughout development.

Project Description

Main Goals

The main goal of the Initiative Table is to provide a visual interface that manages several aspects of the BETA Universe Systems game, most notably keeping track of whose turn it is at any given moment, and guiding players through the combat loop. Outside of the game, users are able to create an account with the system, and can make and manage characters and sessions.

At this stage of the project, the focus is on implementing a basic version of the combat system as well as implementing all the other actions users can take while in a session (eg. MOVE, WAIT, HOLD). The combat system requires the use of other systems that need to be developed as well, including adding databases for weapons, armor, and shields, an expanded real-time connection, and the special GM-only features.

Intended Users

The BETA Universe Systems game is meant to be played by a general audience, of all ages and levels of expertise. Normally, players are expected to have some understanding of the rules and processes of the game, which may vary wildly from session to session. Through the Initiative Table app, many of these rules are abstracted away into the user interface, making the game accessible to newer or younger players. Experienced players may also benefit from this system as it allows for custom characters, weapons, and skills to be used in whatever setting the session host deems fit.

Main Functionality and Capabilities

The Initiative Table app provides a host of quality-of-life features that handle many of the administrative and organizational aspects of the BETA Universe System. Through the app, players can create sessions and invite other players to play in real-time. Within each session, the app will manage several aspects of play, the most notable of which is the Initiative Table, a system that determines whose turn it is at any given point in the game.

The app will make use of both a web and mobile interface. A database and login system will be implemented so that players can save their characters, weapons, armor, and shields to be used across various sessions. Features like computer-generated dice rolls and automatic calculation of attack damage, stat changes, and initiatives will also be provided.

Technologies Used

The application's front-end will be developed using Vue.js, an open-source front-end JavaScript framework, alongside Vuetify, which provides a library of pre-made UI components. To communicate with the backend, we will use the Axios library. Our front-end languages are HTML, CSS, and JavaScript.

The back-end of our application will consist of Microsoft's ASP.NET Core framework. Our data will be saved to a Microsoft SQL Database. We will be using Entity Framework as our ORM to manage the data in the database. To generate real-time connections, we will be using the SignalR library. The back-end is written entirely in C#.

Dependability

The nature of the game does not require that campaigns must be played in one sitting. Therefore, to ensure the reliability of the service, session states will be preserved until the room is deleted, meaning that players can jump right into a game at any time and continue right where they left off.

While the app doesn't make use of any information more sensitive than an email, there are still some measures taken to provide safety and security. When registering an account with the system, the user is prompted to enter in a code that is sent to the email address they provided. User passwords are also hashed before being stored in the database. The application also limits certain data each user is allowed to view and modify.

Significance

The domain of tabletop RPGs has been around for decades but has not been the focus of technological development as much as other entertainment industries have seen. This makes it incredibly interesting to see what possibilities surround this project and how technology will affect the evolution of gameplay.

Professionally, the team will be exposed to many technologies that were not familiar. This will force the expansion of the team's technical expertise and individual toolsets. The highly integrated nature of the app components will also push the team to higher standards of communication and cohesion. Finally, the project itself will become an academic achievement that will help launch the team's post-graduation careers.

The project's primary focus is new and innovative because it addresses such a niche and abstract problem that does not have much in existing technology or research. This application is known as "Theme Agnostic," meaning that any game theme desired by the players can be supported by this application, which is an innovative change to the tabletop RPG industry.

Few other applications attempt to provide the same set of services as our project; however, there are systems within the same domain that provide services in various ways. One such web application is Roll20 (The Orr Group, LLC 2022). This application hosts web-based virtual tabletop games with a finite list of pre-existing games. Another program in the same realm is PCGen (PCGen Project 2022), which is specific to character creation for tabletop RPGs, but again, is specific to pre-existing game themes. Finally, D&D Beyond is a character creation service as well as a rules compendium for D&D (D&D Beyond 2022).

The market value for new tabletop RPGs is staggering. The industry and its popularity have significantly grown with the pandemic. With the growing presence of tabletop RPGs in mainstream media, for example, in Stranger Things, the genre has become more popular than it once was. It is a versatile system that will allow any genre of play with the same rules.

This project solution is a part of a suite of other applications that merge to make the role-playing game by BETA Universe Systems more efficient and enjoyable. Due to this project being a part of a suite, the enhancements can be a connection to the overall suite and a more streamlined UI that will have the ability to send that information to other suite applications.

The social impacts of this type of technology has the potential to become obvious and widespread. One of the limiting factors for this type of gameplay is the need to be in the same physical location with other players for long periods of time. Much of this is due to the lengthy calculations our project aims to eliminate, thus making gameplay much faster and more enjoyable, increasing the positive social impact of the entire gaming community.

Environmentally, the digitization of much of the game's physical components can have a positive effect on the use of stationary materials. There is also a minimal amount of cloud data usage and will have a negligible impact on the usage of natural resources associated with cloud computing.

Legal and Ethical Aspects

There are no potential legal issues since we are working to complement an original RPG tabletop system. We are using technology we have built ourselves, or that is open-source. Many separate applications do what ours will be able to do. The difference with ours is we combine all of those features in one package and include a real-time attack system that will differentiate ours from the rest. There is next to no risk of potential legal action against us. We will ensure our product meets the highest professional standards and our advisor's standards by not using code that is not ours, making our code neat and readable, testing our code extensively to ensure it is polished and bug-free. We will also refactor our code wherever possible, meet extensively with our advisor at various points throughout development, and provide extensive design documentation through the development of the project. We are also crediting anyone's code we use in our project. We do not claim anyone's as our own.

Changes and Progress since the Initial Concept

We have managed to finish our main UI, chat system, session system, character system, and authentication system. To be more specific, we have finished the styling of our application's main page and the in-session page. We have completed the central character system components and select screen, where the player can create, edit or choose a character to join a session. While the primary operations of the character system are complete, we still need to save items related to that character, such as shields, armor, and weapons; this will require a few more hours to fully complete. We have finished our session system that allows a GM to manage sessions and players to join real-time sessions with a built-in chat log. In addition, we have finished the authentication system where new people can register for accounts verified with a six-digit code sent to the provided email or log in to existing ones. This data is currently being stored and pulled through our MS SQL database. Our combat system currently has only minor features implemented, requiring additional work.

Since October, we have had to dial back our goals to more realistic ones. We will be implementing what we can in the given time. More specifically, we will be cutting back on the additional quality of life features we meant to implement in our combat system. Instead, we will focus on the basics and work on the remaining other features if time permits. The main reason for this change is that the real-time session technology has proven to be difficult to debug, and for that reason, it will take longer to develop features for it. This meets our advisor's satisfaction as well as ours. We are all very happy with the progress we have made so far. Due to this, we have put those off for the time being, and if time permits, we will complete them.

Project Responsibilities

The BETA Universe Systems Initiative Table Application is divided into four main components: the character profile component, the authentication component, the session management component, and the real-time session component. The character profile component allows the creation and management of character profiles. The authentication component enables players to register and log into their accounts. The session management component will enable players to join sessions with their selected characters and the Gamemaster to manage them. The real-time session component includes a chat log and our combat loop.

To ensure an even distribution of work and responsibility, these four components were subdivided into two subcomponents that each team member was responsible for completing from concept to implementation and testing. These two subcomponents were the front-end and back-end for the component. The front-end of each system involved client-side code written in HTML, CSS, and JS. The back-end of each system involved server-side code written in C#. The following table demonstrates how these components were divided and which team member was responsible for it.

Component	Subcomponents	Team members accountable
Character profile	Front-end Back-end	Mark Jacob
Authentication	Front-end Back-end	Griffin Andy
Session management	Front-end Back-end	Jacob Jacob
Real-time session	Front-end Back-end	Jake Andy

Table 1. The table above lists the division of each component and the team member responsible.

While some team members were responsible for more subcomponents than others. The work was evenly distributed based on the size of the workload and the importance of each subcomponent. In addition, multiple team members worked on each component in addition to what is shown in the table above.

Project Monitoring and Risks

Outline

The plan to monitor the team's progress and to make sure that the project gets done on time is through application-aid, trello and discord for communications and task management, as well as peer responsibility. This way the team has an ability to stay in contact with each other to communicate problems and get assistance if bugs arise or a subject/ feature is outside the purview of a members' capability. Additionally, the team leader has set out both soft and hard deadlines so that the team members know when a feature/ prospect is to be done. If there is the possibility of a member falling behind or not completing their tasks on the expected deadline, then the team leader will get in contact and reassess the deadline to find out what the latest date can be set for project completion and feature/prospect completion.

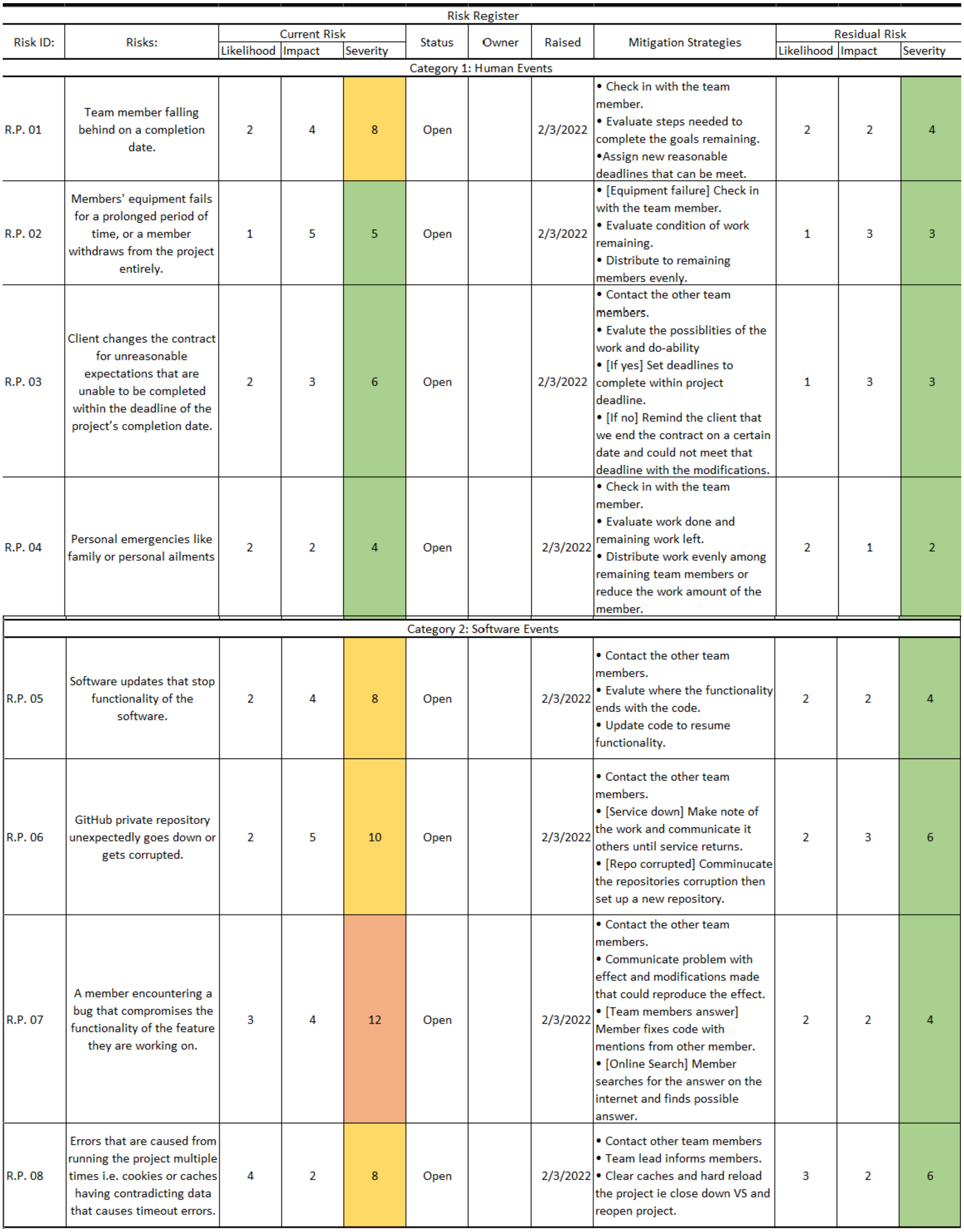
Risks

The following are risks that can occur with the project in terms of the project not meeting the completion deadline. The goals

1. Team Member falling behind on a completion date.
    a. Check in with the team member.
    b. Evaluate steps needed to complete the goals remaining.
    c. Assign new reasonable deadlines that can be met.
2. A member encounters a bug that compromises the functionality of the feature they are working on.
    a. Contact the other team members.
    b. Communicate problem with effect and modifications made that could reproduce the effect.
    c. [Team members answer] Member fixes code with mentions from other member.
    d. [Online Search] Member searches for the answer on the internet and finds a possible answer.

3. Errors that are caused by running the project multiple times i.e. cookies or caches having contradicting data that cause timeout errors.
    a. Contact other team members
    b. Team lead informs members.
    c. Clear caches and hard reload the project i.e. close down VS and reopen project.
4. Personal emergencies like family or personal ailments
    a. Check in with the team member.
    b. Evaluate work done and remaining work left.
    c. Distribute work evenly among remaining team members or reduce the work amount of the member.
5. Software updates that stop the functionality of the software.
    a. Contact the other team members.
    b. Evaluate where the functionality ends with the code.
    c. Update code to resume functionality.
6. The client changes the contract for unreasonable expectations that are unable to be completed within the deadline of the project's completion date.
    a. Contact the other team members.
    b. Evaluate the possibilities of the work and do-ability
    c. [If yes] Set deadlines to complete within project deadline.
    d. [If no] Remind the client that we end the contract on a certain date and could not meet that deadline with the modifications.
7. GitHub's private repository unexpectedly goes down or gets corrupted.
    a. Contact the other team members.
    b. [Service down] Make note of the work and communicate it to others until service returns.
    c. [Repo corrupted] Communicate the repository's corruption then set up a new repository.
8. Members' equipment fails for a prolonged period of time, or a member withdraws from the project entirely.
    a. [Equipment failure] Check in with the team member.
    b. Evaluate the condition of work remaining.
    c. Distribute to remaining members evenly.

Risk Register



Risk Register											
Risk ID:	Risks:	Current Risk			Status	Owner	Raised	Mitigation Strategies	Residual Risk		
		Likelihood	Impact	Severity					Likelihood	Impact	Severity
Category 1: Human Events											
R.P. 01	Team member falling behind on a completion date.	2	4	8	Open		2/3/2022	• Check in with the team member. • Evaluate steps needed to complete the goals remaining. •Assign new reasonable deadlines that can be meet.	2	2	4
R.P. 02	Members' equipment fails for a prolonged period of time, or a member withdraws from the project entirely.	1	5	5	Open		2/3/2022	• [Equipment failure] Check in with the team member. • Evaluate condition of work remaining. • Distribute to remaining members evenly.	1	3	3
R.P. 03	Client changes the contract for unreasonable expectations that are unable to be completed within the deadline of the project's completion date.	2	3	6	Open		2/3/2022	• Contact the other team members. • Evalute the possiblities of the work and do-ability • [If yes] Set deadlines to complete within project deadline. • [If no] Remind the client that we end the contract on a certain date and could not meet that deadline with the modifications.	1	3	3
R.P. 04	Personal emergencies like family or personal ailments	2	2	4	Open		2/3/2022	• Check in with the team member. • Evaluate work done and remaining work left. • Distribute work evenly among remaining team members or reduce the work amount of the member.	2	1	2
Category 2: Software Events											
R.P. 05	Software updates that stop functionality of the software.	2	4	8	Open		2/3/2022	• Contact the other team members. • Evalute where the functionality ends with the code. • Update code to resume functionality.	2	2	4
R.P. 06	GitHub private repository unexpectedly goes down or gets corrupted.	2	5	10	Open		2/3/2022	• Contact the other team members. • [Service down] Make note of the work and communicate it others until service returns. • [Repo corrupted] Comminucate the repositories corruption then set up a new repository.	2	3	6
R.P. 07	A member encountering a bug that compromises the functionality of the feature they are working on.	3	4	12	Open		2/3/2022	• Contact the other team members. • Communicate problem with effect and modifications made that could reproduce the effect. • [Team members answer] Member fixes code with mentions from other member. • [Online Search] Member searches for the answer on the internet and finds possible answer.	2	2	4
R.P. 08	Errors that are caused from running the project multiple times i.e. cookies or caches having contradicting data that causes timeout errors.	4	2	8	Open		2/3/2022	• Contact other team members • Team lead informs members. • Clear caches and hard reload the project ie close down VS and reopen project.	3	2	6

Fig 1. This figure is the risk register of the risk for the project of team 7.

Work Contribution

	Andy Alarcon	Jacob Gayban	Mark Graham	Jacob Tucker	Griffin Wagenknecht
Project Assignment 1 Paper (Writing sections and formatting)	2.0	2.5	2.0	2.0	2.0
Winter Break work (Includes coding and other planning)	4.0	0.0	0.0	0.0	6.0
Researching Web Development	0.0	1.0	1.0	1.0	0.0
Meeting with teaching team for project progress	0.5	0.5	0.5	0.5	0.5
Team meetings	2.0	2.0	2.0	2.0	2.0
Total	8.5	6.0	5.5	5.5	10.5

Table 2. The table above shows the amount of time spent by each team member on each activity.

References

D&D Beyond 2022, accessed 1 February 2022, <https://dndbeyond.com/>

PCGen 2022, accessed 1 February 2022, <http://pcgen.org/>

Roll20 2022, The Orr group, LLC, accessed 1 February 2022, <https://roll20.net/>