

University of Nevada, Reno
Department of Computer Science and Engineering

Project Title: Guild

Team #21

Riley Moore

Blake Cash

Ryan Van

Instructors:

Sergiu Dascalu

Devrin Lee

External Advisors:

Chase Carthen

Mitchell Martinez

February 22, 2019

Table of Contents

Abstract.....3

Recent Project Changes.....3

Updated Specification.....3

 Summary of changes in project specification.....3

 Updated technical requirements specification.....4

 Updated use case modeling.....6

Updated Design.....10

 Summary of changes in project design.....10

 Updated high-level and medium-level design.....10

 Updated user interface design.....21

Glossary.....30

Engineering Standards and/or
Technologies.....31

References.....32

Team
Contributions.....33

Abstract

Guild is an event planning app for the iPhone and Android with a strict focus on tabletop gaming, collectible card games, and LAN gaming. In addition to a traditional event planning interface, Guild will provide a large generalized tool set to users and event organizers, helping to facilitate every step of organizing an event, from initial planning to actually running the event. In this way, Guild intends to be considered a 'go-to' application as far as creating easy to organize, effectively run events that are enjoyable for both attendants and organizers. As of now, Guild is a mostly-offline set of tools for facilitating in-person gaming events, with intention to expand into an online social service as well.

Recent Project Changes

Since writing project part one two weeks ago, the major changes to guild have occurred in the toolkit section of the application as well as in laying out Guild's database schema. Since our last report, with Chase's assistance, Ryan has laid out what is considered to be very close to a finalized database schema for Guild, which will likely need minor updates going forward but is considered mostly complete as far as Guild's current intended feature list is concerned. Meanwhile, Riley, Blake and Mitchell have gone about creating Guild's initial toolkit functionality and working to make the pieces robust enough for use during real events. Riley and Blake have also taken Guild to an actual in-person gaming event and used some of the tools they've created there to test them and get feedback.

Updated Specification

Summary of changes in project specification

Unlike what you might expect from a project of this scale, Guild's intended feature list has not shrunk at all since the completion of project part 2 in CS425. On the contrary, some good ideas brought up by our stakeholders have actually ended up being added to the hopeful feature list because the team liked them so much. As you'll see on the technical requirements specification, the list of requirements has grown rather than shrunk, though all added requirements have been given a low-priority 'level 3' designation, as the team really enjoys the ideas, but doesn't think that it's reasonable to expect to have them completely done in the time frame given. This is partly due to the team's philosophy discussed in earlier project reports, where the team would rather have a large, ambitious feature list and not get to every single feature, rather than have a very small list of features that leaves them with a boring final product.

Updated technical requirements specification

Functional Requirements

FR01	1	Allow users to create and manage events, to which the user may invite other users to join.
FR02	1	Group chat functionality for members of groups or events.
FR03	1	Ability to create of private or public groups of users.
FR04	1	Creation of a calendar for each user which they may edit and add events to.
FR05	1	Ability to search for and register for public events that other users have published to Guild.
FR06	1	Ability to search for and invite players to both public and private events
FR07	1	Functional tools for use during events including dice rollers, note passing, chat channels, scorekeepers, and current bracket progression.
FR08	2	Ability to upload multiple data types, including decklists, character sheets, and brackets.
FR09	2	Ability to create a profile that details what games you play and other player info, characters played
FR10	3	Web scraping implemented to automatically update user profiles from other sources with information such as tournament placings and deck lists.
FR11	3	Player profiles that automatically aggregate tournament results and character/deck choice to create a more complete profile with no user input
FR12	3	'Turn-by-turn' smart interface that stays constant for a full group, giving who's turn it is as well as information like active buffs, armor class.

Table 1: Functional Requirements

Non-Functional Requirements

NFR01	1	Distribute Application on two target platforms, IOS and Android devices.
NFR02	1	Token generation security system.
NFR03	2	Application should load several pages within 5 seconds, which the user can then navigate with no load time.
NFR04	3	Desktop application created.
NFR05	3	User-generated content retrieved in under 5 seconds when requested.

Table 2: Non-Functional Requirements

Updated use case modeling

Use Case Diagram

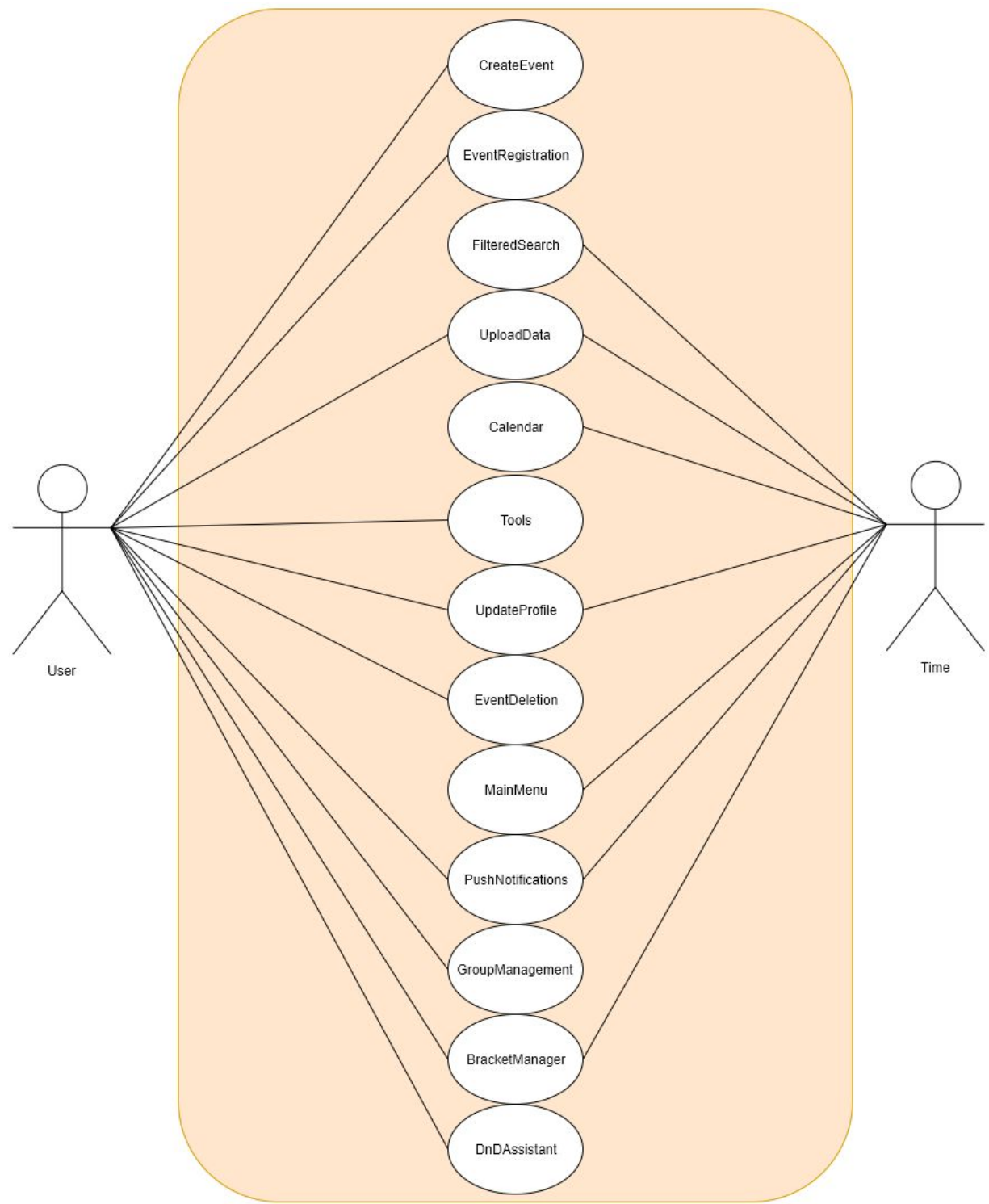


Figure 1: Use Case Diagram

Use Case Descriptions

ID	Use Case	Description
UC01	CreateEvent	User will be able to set the date, time, location, game type, and general information of their event. User/users who create an event will have admin permissions, which enable them to manage their event all the way from beginning to end. Admins can either send it to the database as a public event or as a private event which only certain users or groups can see.
UC02	EventRegistration	Users will be able to register for an event listed on the applications and have it propagate onto their calendar and notifications. A user may register into an event in two ways: by accepting an invitation to either a public or private event, or by selecting a public event after having searched one out and registering.
UC03	FilteredSearch	User will be able to select specialized categorizes to minimize the amount of items they are searching for. Filters for game type, location, and minimum host ratings will be available for the user to enable or disable when looking for an event. Users may search for both players and events through this system
UC04	UploadData	The user will be able to upload files necessary for the running of events and have them hosted on the database. Users may also upload their google calendar so that groups may coordinate events when users are most available.
UC05	Calendar	User will be able to view all the events that they have registered for based on a month, week, day format. On the day format, the events will be listed by the hour it starts and ends.
UC06	Tools	User will have access to a variety of tools to assist with their tabletop and LAN gaming experience. (i.e dice roller, deck list, tournament brackets, etc.) At the main menu, a user will select the tools options which will navigate them into a subsection of the

		application with more specific features. For example, if a user wants to access a specific dice for Dungeons and Dragons, he will navigate through tools > tabletop > dice rollers > D20.
UC07	UpdateProfile	Users will be able to provide additional information regarding what their interest are related to the platform. (i.e favorite board game, prefer being Dungeon Master, etc.)
UC08	EventDeletion	User will be able to delete an event that they have created. Only the admins or the creator of the event will have the permissions to delete the event.
UC09	MainMenu	The app will have three main options after logging in. The user may select Events, Profile, or Tools, which will all bring the user to a new set commands specific to that selection.
UC10	PushNotifications	If a user has listed themselves for an event, a notification will alert the user a certain amount of time before to remind them of the event. The user can tap this notification to link them to the event page within the app. Push notifications may also be enabled to see when a user has been messaged.
UC11	GroupManagement	A user may navigate to the group creation menu from the main menu, invite other users to a private or public group. Users in the group may create and share events and also chat amongst each other.
UC12	BracketManager	Admins will be able to create, edit, and post tournament brackets to the database, while regular users will be able to view the bracket and upcoming matches. Brackets will also keep track of character and deck choices and use this data to update user profile information.
UC13	DnDAssistant	Using a predetermined group of players, the application will allow users to cycle through each player based on their given initiative score and display relevant stats like armor class and health.

Table 3: Detailed Use Cases

Traceability Matrix

		Use Cases												
		UC 01	UC 02	UC 03	UC 04	UC 05	UC 06	UC 07	UC 08	UC 09	UC 10	UC 11	UC12	UC13
Requirements	R 01	X							X	X				
	R 02										X	X		
	R 03									X		X		
	R 04					X				X				
	R 05		X	X						X				
	R 06			X						X				
	R 07						X			X			X	X
	R 08				X								X	
	R 09						X	X		X				
	R 10							X					X	
	R 11				X								X	
	R 12													X

Table 4: Requirement Traceability Matrix

Updated Design

Summary of changes in project design

Guild's general project design remains much the same as it has since the application's inception. General user interface flow remains much the same despite UI updates streamlining the process on the front end. The app will continue to be put into two branches, the event planning and toolkit. The most significant updates to project design come on the back-end, where database schema has been updated and finalized as detailed below in accordance with the team's goals for the application.

Updated high-level and medium-level design

System-level Diagram

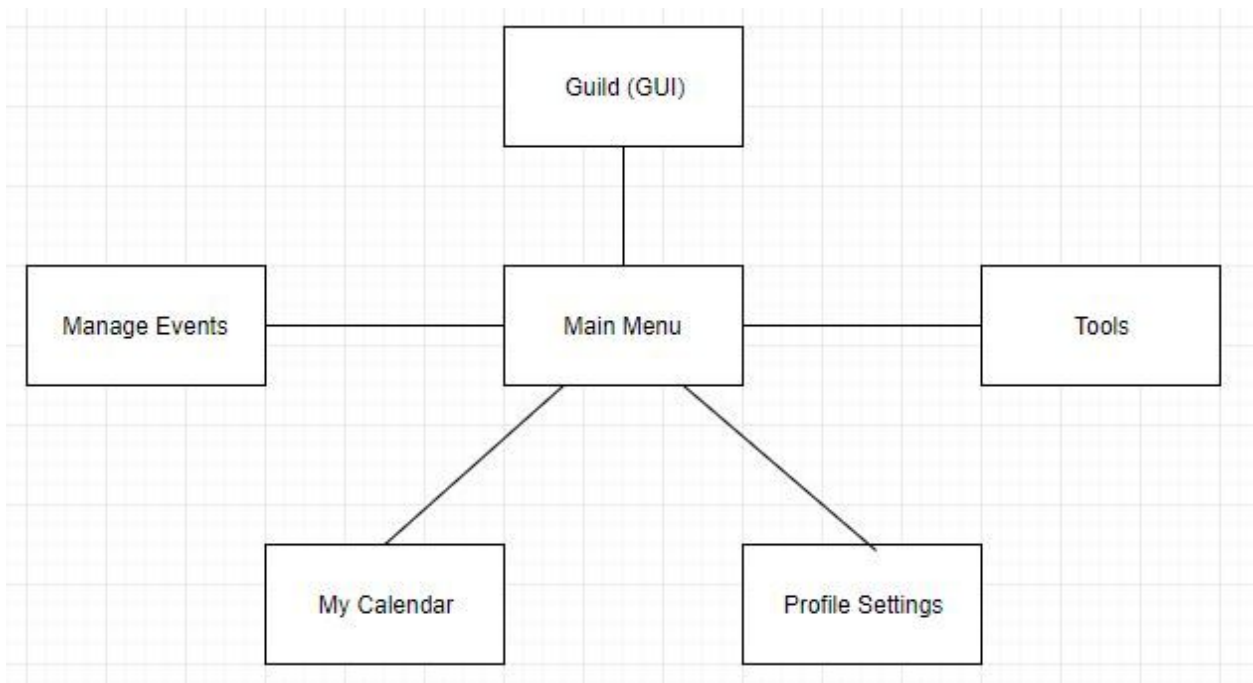


Figure 2: Guild Context Model

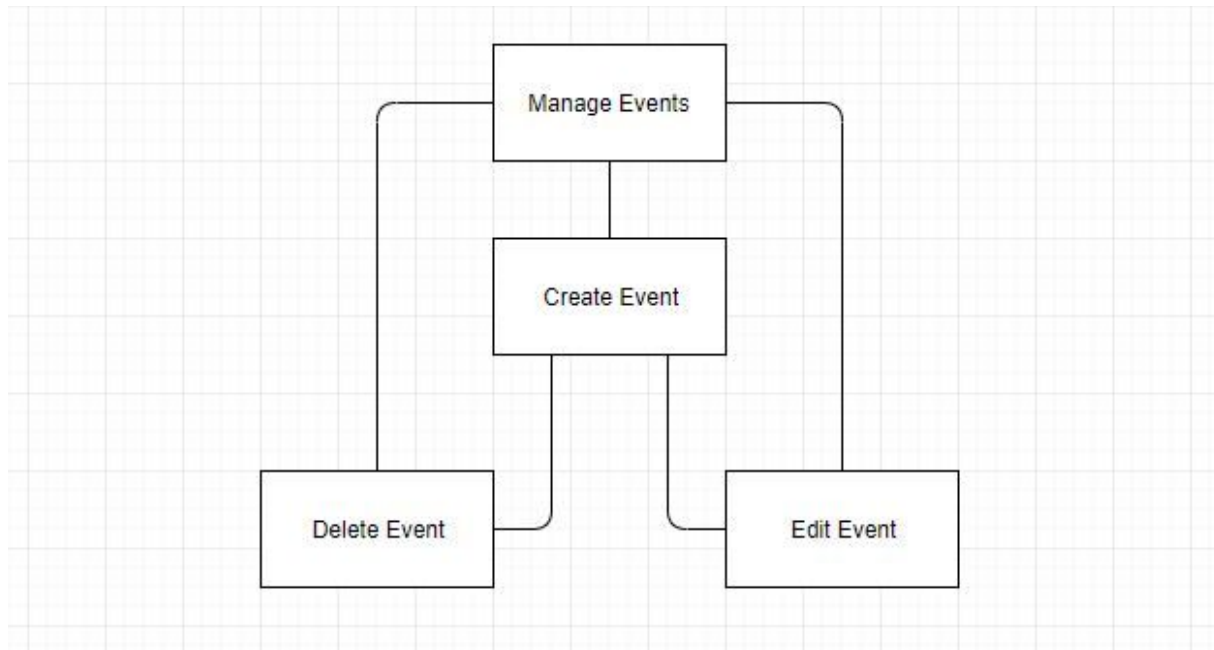


Figure 3: Manage Events Context Model

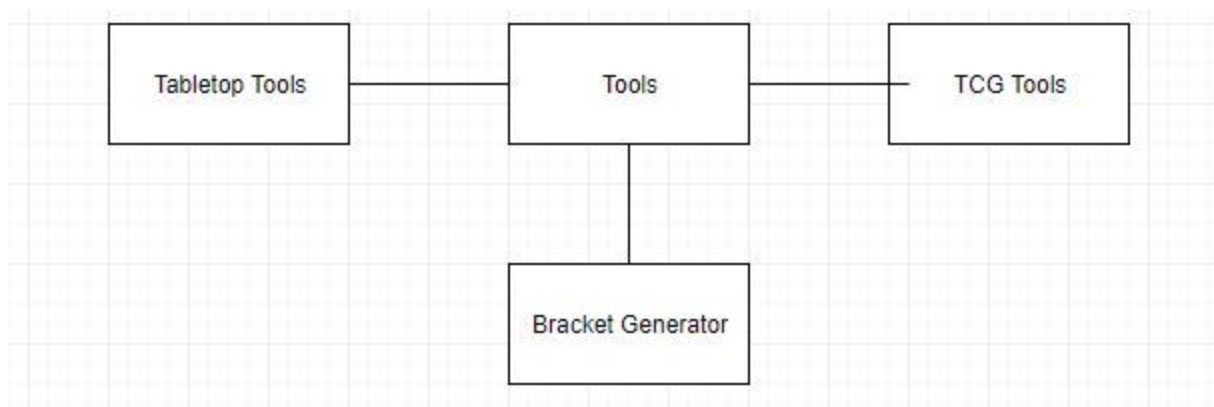


Figure 4: Tools Context Model

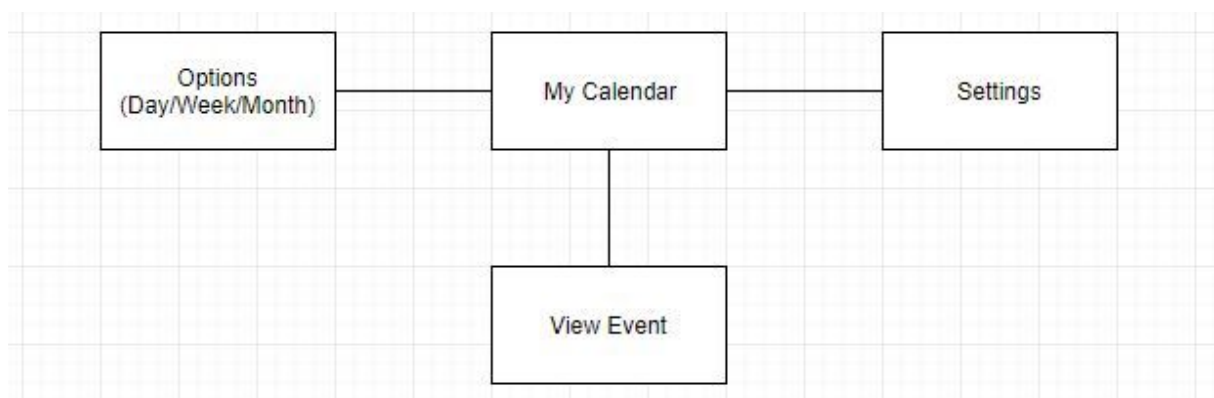


Figure 5: Calendar Context Model

Class Diagram

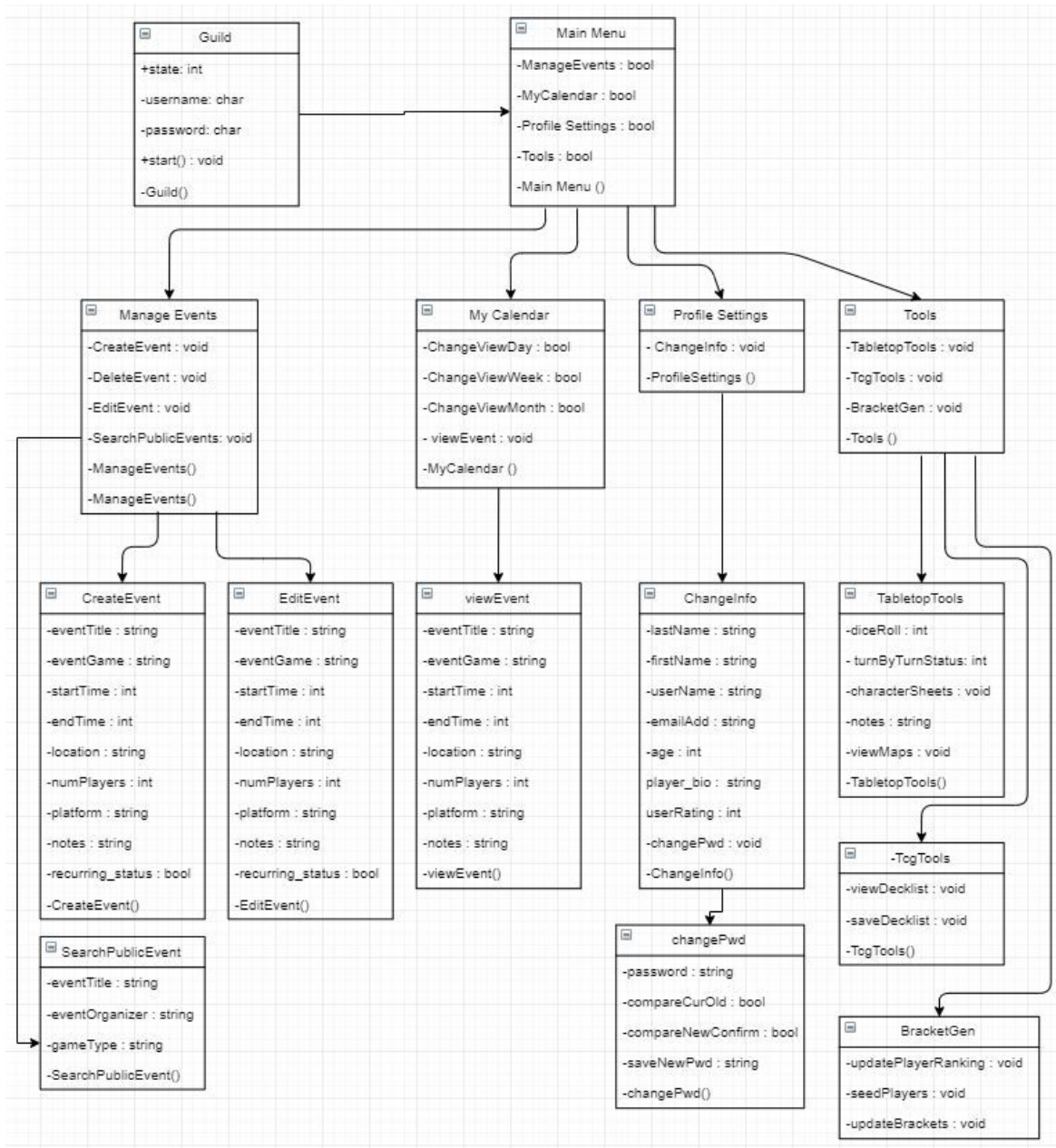


Figure 6: The design class diagram of Guild

Program Units

Guild

This the class is the starting point of the Guild application. The user will be required to enter their username and password to continue forward into the app. The start function is what confirms credentials are correct.

Main Menu

The main menu class is used to display Guild's functions and tools for the user.

Manage Events

The Manage Events class gives the options to create, edit or delete an event. Once a option has been selected, they'll be taken to the selected item.

CreateEvent

The user will be given many item or variables to fill out in order to create the event. All the data entered will be placed into the Events database.

EditEvent

The EditEvent class will allow the user to modify any of a pre-existing event information that they have created and ownership of.

SearchPublicEvent

The SearchPublicEvent class will allow the user to search for any public events using specific criterias; event title, event creator, and game genre. A list of events related to the information provided will propagate on the display once a search has been done.

My Calendar

The My Calendar class will give the user options to change to way they view the calendar and view any events that they have joined. The options of views consist of a day, week, and month.

ViewEvent

In the ViewEvent class, the viewer will be able to see all the event details that have been inputted by the creator of the event.

Profile Settings

The user will be able to view their current profile. They have the option to change any information on their profile or account if need be.

ChangeInfo

The user can change any information regarding their profile, such as their email and player bio.

changePwd

The changePwd class will allow the user to change their password. The requirements to do so will be that they are able to enter their old password correctly, and able to confirm they entered there new password correctly.

Tools

In the Tools class, the user will be able to select which type of tools they'll be using to improve their gaming experience.

TabletopTools

The TabletopTools provide a few tools to help those who enjoy playing tabletop games such as Dungeon and Dragons. Some of the tools include a dice roller and a way to determine whose turn it is to play in that round.

TcgTools

The TcgTools provide a way for users to save or view decklists that have been uploaded onto Guild.

UserInfo		
Primary Key	ID <int>	The key used to properly identify the correct user
	LastName <varchar>	User's last name
	FirstName <varchar>	User's first name
	UserName <varchar>	User's username for accessing Guild and will function as their display name
	Password <varchar>	User's Password for accessing Guild
	Email <varchar>	User's email address or logon credential for accessing Guild
	Age <int>	User's age
	Player_Bio <varchar>	The description or bio that the player enters about themselves
Foreign Key	User_Rating <int>	The user's rating status based on reviews of other users in Guild. This cannot be edited by the user.

Table 5: Database table for user information

EventInfo		
Primary Key	ID <int>	Key used to identify the proper event
Foreign Key	UserInfo	The creator and admin of the event
	Event_Title <varchar>	The title used for the event
Foreign Key	GameInfo <varchar>	The game that will be played at the event
	Start_date <varchar>	The date and time of when the event will start
	End_date <varchar>	The date and time of when the event will end
	Num_players <int>	The number of player needed to start the event

Foreign Key	Platform <varchar>	What type of platform will the event be played on (PS4, Xbox One, tabletop, etc)
Foreign Key	Notes <varchar>	Any additional information that needed to be added to the event
	Public <boolean>	Is this event listed publicly or only for those invited

Table 6: Database table for events created

GameInfo		
Primary Key	ID <int>	Key to identify the correct game, assuming there exist games with very similar names
	Game_Title <varchar>	The proper title of the game
Foreign Key	Platforms <varchar>	What platform can the game be ran on (PS4, PC, etc)

Table 7: Database table for the games shown in Guild

Location		
Primary Key	ID <int>	Key to identify the correct Location
Foreign Key	EventInfo <varchar>	Foreign Key to the EventInfo Table
	Location_Name <varchar>	Name of the location
	Country <varchar>	Country's name
	State <varchar>	State's name
	Address <varchar>	Address of the location
	Zipcode <int>	Zip Code of the location
Foreign Key	Notes <varchar>	Foreign Key to Notes table

Table 8: Database table for location information

Platform		
Primary Key	ID <int>	Identify the Platform data used for game/event
	Type <varchar>	The type of platform (PC, PS4, XBONE, etc)

Table 9: Database table for game platform information

UserRating		
Primary Key	ID <int>	Identify the correct rating
	User_ID <int>	The ID of the user who created the rating
	Rating <int>	The range of the rating (i.e 1 - 5 stars)
Foreign Key	Notes <varchar>	Foreign Key to Notes Table

Table 10: Database table for user rating information

Notes		
Primary Key	ID <int>	Identify the proper notes to be used
	Subject <varchar>	The category the note falls under. The table name that has Notes as a foreign key
	Description <varchar>	The information the user desires to input
	Date <varchar>	Date of when the note was created
	Time <int>	Time the note was created

Table 11: Database table for extraneous information

GroupInfo		
Primary Key	ID <int>	Identify the correct Group
	Name <varchar>	Name of the Group
Foreign Key	UserInfo <varchar>	Foreign Key to UserInfo table
	Group Description	Description of the group

Table 12: Database table for information related to groups

AttendeeList		
Primary Key	ID <int>	Identify the correct attendee list
Foreign Key	UserInfo	Foreign Key to UserInfo
	Status	For those who were invited. Is the user going or still awaiting response
Foreign Key	EventInfo	Foreign Key to EventInfo

Table 13: Database table for storing event attendee information

Notification		
Primary Key	ID <int>	Identify the correct Notification
	Message <varchar>	The message sent to the users
Foreign Key	EventInfo <varchar>	Foreign Key to EventInfo

Table 14: Table for storing information regarding notifications.

Recurring_admin		
Primary Key	ID <int>	Identify the correct Recurring_admin status
Foreign Key	UserInfo <varchar>	Foreign Key to UserInfo table
Foreign Key	EventInfo <varchar>	Foreign Key to EventInfo table
Foreign Key	AttendeeList <varchar>	Foreign Key to AttendeeList table
	Timeframe <varchar>	Determines if the event will be daily, weekly, monthly, etc.
Foreign Key	Notification <varchar>	Foreign Key to Notification table

Table 15: Database information used for recurring events

Decklist		
Primary Key	ID <int>	Identify the correct Decklist
Foreign Key	UserInfo <varchar>	Foreign Key to UserInfo
	Name <varchar>	Name of the deck
	Type <varchar>	Type of the deck (i.e burn, etc)
	Description <varchar>	Describes what the deck is about or contains based on the user's input

Table 16: Database table for storing card game deck related information

Updated user interface design

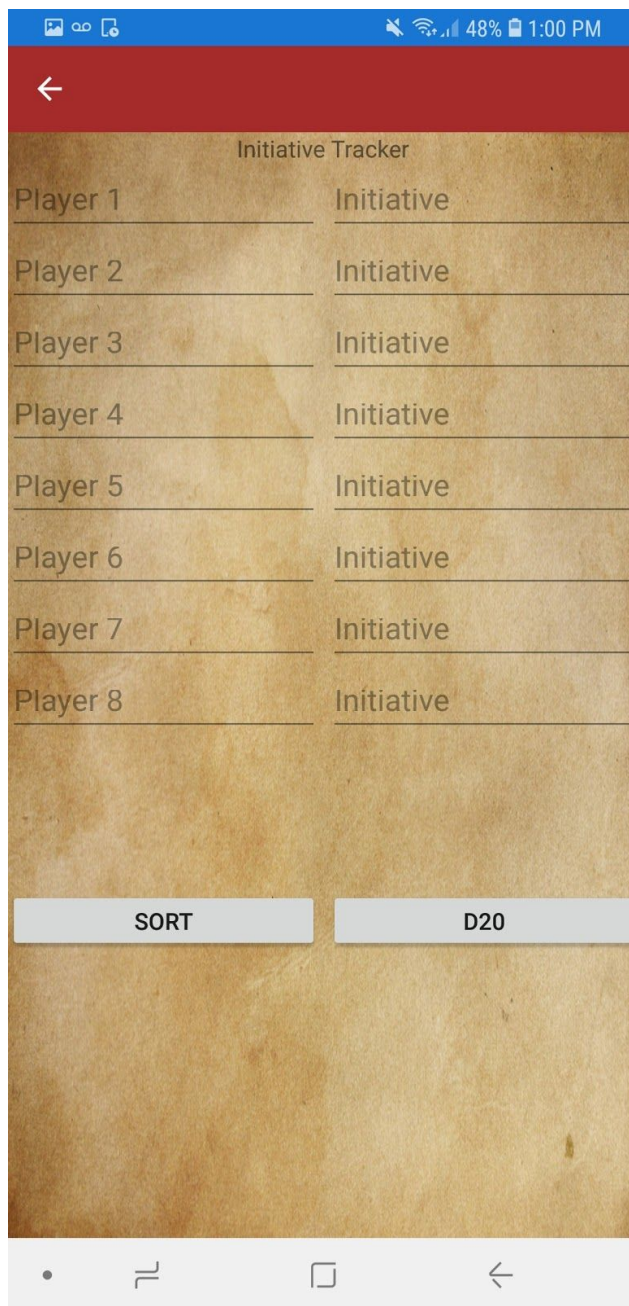


Figure 8: A screenshot of the initiative tracker for Dungeons and Dragons players. the application will take names and initiative scores in the entry fields and upon sorting will then display which user turn it is. The D20 dice is present for the case that it is useful in determining initiative. Modifiers and other calculations to this score will be added in the future.

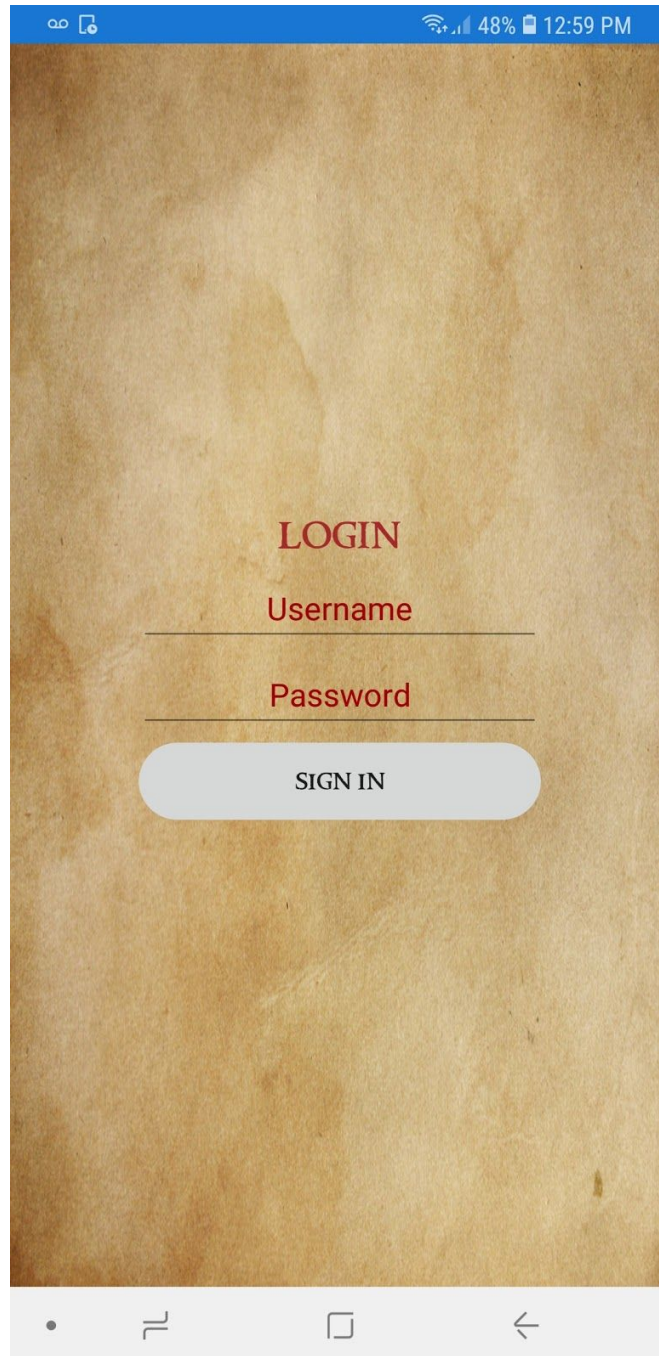


Figure 9: The login screen for guild, currently requires the user to input something in order to get past the empty field catch but does not require registered email or username. Screen will be updated with a remember me option as well as a register button.

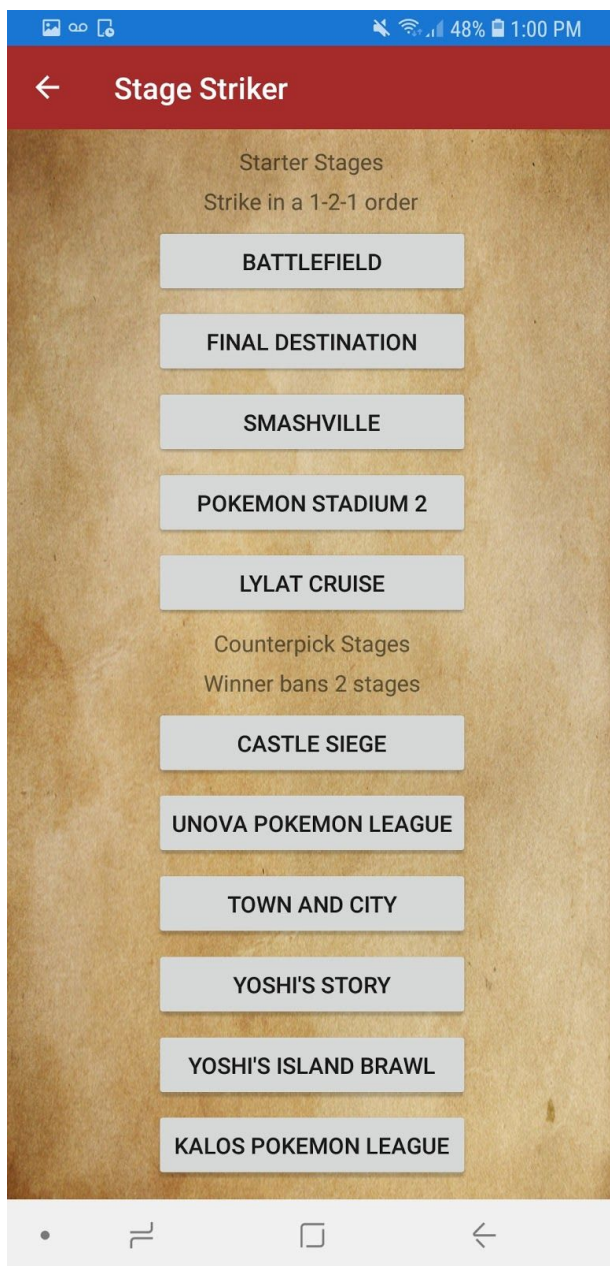


Figure 10: This page is designed specifically for use in Smash Brothers tournaments, which Riley and Blake tested in a tournament on February. Players take turns striking stages from the list until one stage is left, which the players will then compete on.

Currently it is based on the standard competitive ruleset but custom ruleset implementation will be added as well.



Figure 11: The first screen users will see when they log in, this will serve as the main hub for all the features of Guild. Potential changes include a side menu, grid layout instead of a single column of buttons.

EVENT CREATION

Enter Event Name

Enter Event Description

2/22/2019

Start Time

End Time

CREATE EVENT

Figure 12: Event Creation page interface in which users will be able to create events and post them publicly or share them to specific users. Event input fields have not been finalized as of yet and this page is subject to change.



Figure 13: A screenshot of the “My Events” page, shows events that the user is either registered for or hosting. Updates in the future will include logos which represent game types or event types, color coding, and images of the event.

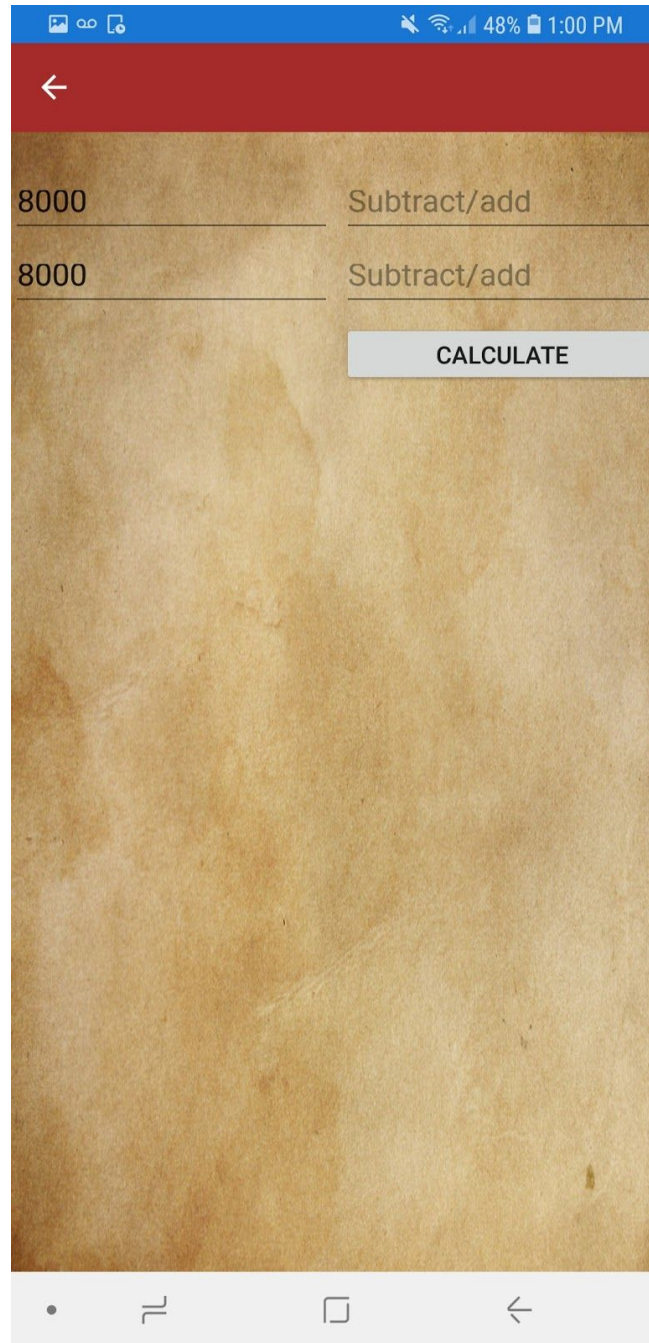


Figure 14: A work in progress screenshot of the card game calculation tool, currently supports keeping track of life totals and modifying them as necessary. Potential improvements include multiple different game support and token trackers.

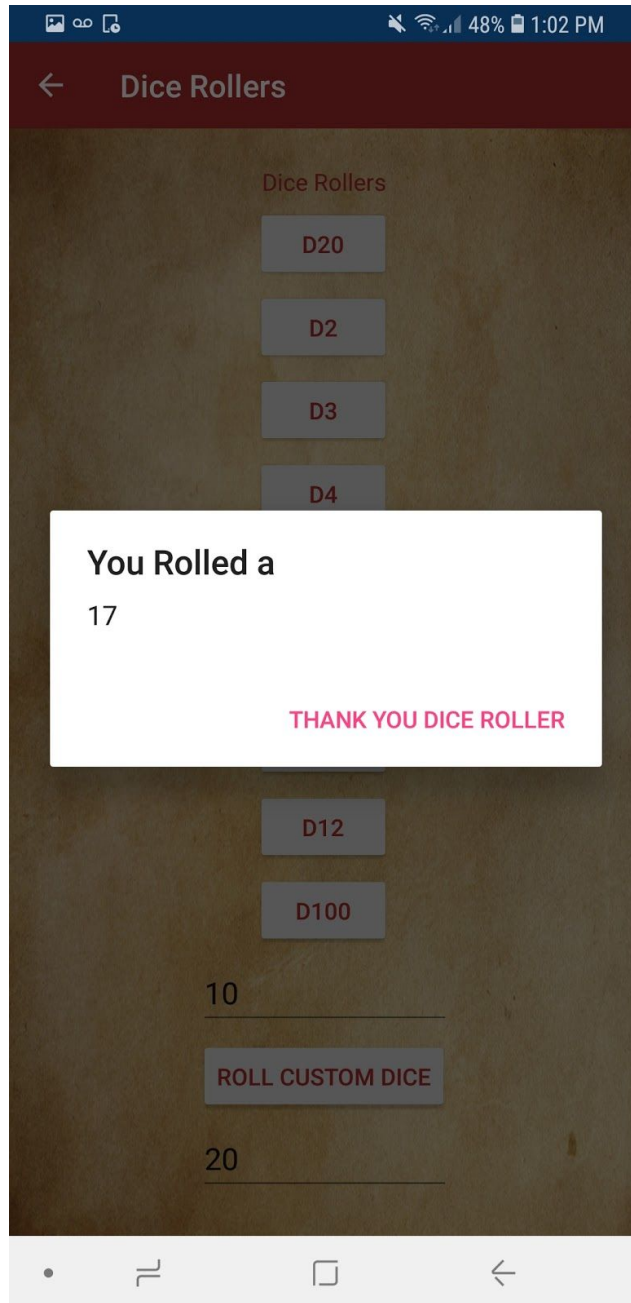


Figure 15: A screenshot of the dice roller page. Fully functioning as intended but needs to be updated aesthetically. Supports most standard dice types in addition to allowing the user to roll a number between any given numbers.

Find Public Events

ENTER SEARCH CRITERIA

Enter Event Name

Enter Organizer Name

Enter Game Type

2/22/2019

2/22/2019

SEARCH

Figure 16: The event search system with which users will input information on what type of events they want to attend. The app will then run a search through the database and return a page that looks similar to the my events page with a list of relevant events that the user can then register for.

Glossary

1. **API:** An Application Program Interface determines how components of software interact based on the set of instructions used.
2. **Database:** Collection of data that is stored and accessed from a computer or server.
3. **C#:** A object-oriented programming language created from Microsoft. A hybrid of C and C++.
4. **Collectible card game(CCG)/Trading card game(TCG):** A strategic card game that uses specially designed playing cards that can be traded between peers.
5. **Cross-Platform:** Represents the fact that the application can be used on both iPhone on Android, and versions of the application on both platforms can communicate with another.
6. **Database Schema:** A representation, depiction, or description of the relationship between the database tables in the database.
7. **Deck List:** A list of all the cards a player is using when playing a strategic card game.
8. **Flask:** A microframework using Python primarily for web development. This will be used to act as the middleman for connecting our database to the mobile application
9. **Forward and Backwards Compatibility:** The concept of an application being able to communicate with older version if the same application (backwards compatibility) or an older version of the application being able to communicate with a newer version (forward compatibility)
10. **Tabletop games:** Games that are usually played on a table or a flat surface. Examples involve board games, card games, and dice games, as well as role playing games like Dungeons and Dragons.
11. **LAN gaming:** A local area network (LAN) is established, allowing those with gaming devices to play multiplayer games within the same vicinity with no delay.
12. **Object Oriented:** A programming model based around 'objects' rather than 'actions' and data over logic. Object oriented programs usually focus on data structures and their interactions
13. **ORM:** Object-relational mapping is a technique used to access a relational database from object oriented languages.
14. **Propagate:** The process of spreading events or information to a wide population.
15. **Python:** A programming language that emphasizes code readability for general-purpose programming.
16. **REST API:** A Representational State Transfer internet protocol that takes advantage of protocols that already exist; Web APIs.

17. **Role-playing game:** A game where players assume a role of a fictional character in a fictional setting.
18. **Smart Phone:** A cellular device that carries functionality similar to that of a computer. Typical smartphones have a touchscreen, and internet access.
19. **SQL:** A Structured Query Language is used to allow communication to the database
20. **SQLAlchemy:** A SQL toolkit and ORM that uses the programming language Python.
21. **SQLite:** An embedded database software for local storage, allowing data to be stored locally rather than on an off-device web server hosting an SQL database.
22. **User Interface:** The way a user and a computer system interact with each other.

Engineering Standards and/or Technologies

(Check SP2019_P2_Prep.pptx on Feb 12 for examples)

Standards:

- UML - Unified Modeling Language is a standard used to creating and documenting artifacts of software systems. The team plans to use UML by constructing the process of a function in Guild along with how the database will be managed; database schema.
- JSON - JavaScript Object Notation is a standard that desires to structure data into a readable format. JSON format will be used by having the converted data inside the database to match the format in Flask. This will allow the front-end side of Guild to process the data efficiently.

Technologies:

- AWS - Amazon Web Services is a on-demand cloud computing platform that provides a variety of infrastructures as a service. The team plans to use AWS by having it host the MySQL database, and Ubuntu VM running the Flask framework.
- Flask - A micro web framework primarily used for web development. Flask will allow communication between the Guild application and the database. The Flask code will be ran in a Ubuntu VM to allow for the code to run constantly.
- Google API - A set of application programming interfaces which allows communication to Google Services. The team plans to use Google credentials to create/access their Guild account. Google Calendar will also be used to within Guild to help manage events the user plans to attend.

- MySQL - A relational database management system based on SQL. MySQL will be used to create, read, update, and delete content in the team's database.
- Xamarin - A mobile application development tool that provides a variety of environments developers can use. Xamarin will provide the ability to build native and/or cross-platform applications. Xamarin will be used to create the UI for Guild along with getting data from the teams Flask server.

References

“Explore Flask Documentation” Journal. Retrieved November 1, 2018 from <https://media.readthedocs.org/pdf/exploreflask/latest/exploreflask.pdf>

- This paper provides in-depth information about the use of Flask, such as best practices when using it. It discusses instances that are commonly appear when using Flask and what is a good approach to that instance. The documentation supports itself by including examples of code and the variable names.

“STUDY ON XAMARIN CROSS-PLATFORM FRAMEWORK” Journal Article. Retrieved October 21, 2018 from

<https://www.ijtra.com/view/study-on-xamarin-cross-platform-framework-.pdf?paper=study-on-xamarin-cross-platform-framework-.pdf>

- This is an academic journal that goes in depth into how exactly Xamarin is useful for cross-platform development. It highlights the advantages of using Xamarin specifically over it's contemporaries and had a hand in helping us decide to use Xamarin for our application.

“SQL” Journal Article. Retrieved October 31, 2018 from

<https://researcher.watson.ibm.com/researcher/files/us-dchamber/SQL-encyclopedia-entry.pdf>

- This journal discusses what SQL is and the purpose of it. SQL is well known for being the most widely-used database query language. It handles data in forms of tables with each table consisting of at least one column. The journal also provides example of code to illustrate how SQL is used to query data.

Contributions of team members

Blake Cash: For this portion of the project, Blake Cash focused on the abstract, recent project changes, and recent updates specification summary, as well as recent design update summary. In addition, he helped update the technical requirements and the use cases for the program. Like all other team members, Blake also helped proofread the assignment. In total, Blake worked for approximately 4 hours.

Riley Moore: For project part #2 Riley handled the UI portion, taking screenshots from the current Guild interface. In addition, Riley also went about updating the use cases and traceability matrix to better match the state of the current project. In total, Riley worked for approximately 4 hours.

Ryan Van: For project part #2, Ryan focused on creating the data structures section in the high-level and medium-level design portion. Ryan also typed the Engineering Standards and/or Technologies portion of this project. He updated the figure for the use case modeling section to match current use cases. Ryan spent approximately 4.5 hours working on this project.