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

March 12, 2019

Table of Contents

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.

Project Updates and Changes

Since writing project part two, a majority of Team #21’s progress has gone into developing flask communication code to ensure the database and application communicate properly, as well as work on bug fixes and visual design updates. This means that between the last paper and now, Guild has mostly implemented it’s event creation code, and work has shifted from what is considered ‘framework’ into refining the existing code and adding features. The list of use cases that guild properly fulfills has not expanded significantly (Aside from the ability to find and create events), but the tools already in place are now notably more robust and usable, compared to their initial implementations which were considered buggy and mostly unintuitive. For Ryan and Chase, work has started on properly moving Guild from it’s initial first-draft database schema onto it’s completed ‘2.0’ database. For Riley, Blake, and Mitchell, time was primarily put into writing C# code for Xamarin that ensured proper database communication, in addition to refining the previously-developed toolkits into less buggy subroutines that would be actually useful in real local gaming settings. As was the case in previous project parts, no notable changes to design or specification of the project have occured, and the overall goals of the project have not changed in any meaningful way.

User Stories and Acceptance Criteria:

Subsystems	User Stories	Acceptance Criteria
Event Planning:	● As a planner, they will be provided with simple categories to fill in order to create an event. ● As a planner, they will be able to make a event public to allow anyone to join the event.	● Simple UI is used to allow for events to be created when clicking of the Create Event button ● Ability to set a event status public or private
Joining Event:	● As a user, they will be able to join a public event or accept an event invitation to be able to participate in the event. ● As a user, they will be able to view details of an event to decide if they desire to participate.	● Ability to search for public events ● Ability to see all details related to selected event
Profile Creation/Edit:	● As a user, they can input information about themselves to allow others to determine if the user would enjoy the event. ● As a user, they can update/change certain information on their profile to show other users.	● User is able to create an account for Guild ● User can go to settings and edit detail on their profile
User Rating:	● As a user, they can provide a review of another user of Guild to notify others of their experience with the specified user. ● As a user, they can view their own rating to see where they performing well or poorly when interacting with others at the event.	● Numeric value is of user's current rating is displayed on the profile ● Able to rate others if both attended the same event
Tool Kit:	● As a tabletop player, they can use the dice roller function to have the capabilities to use dice in a game. ● As a CCG player, they can use the point system implemented in Guild to determine who is in the lead of the match.	● Ability to use majority of tool kits offline ● Displays the tools functions properly

Table 1: Subsystems of Guild outlined with User Stories and Acceptance Criteria

Testing Workflow

Happy Path Workflow 1	**Successfully uploading a public event to the app database.**
Step 1	Select create events from the event manager menu.
Step 2	Fill in the event information on the event creation screen (Name, date, time, location, etc).
Step 3	User finalizes this info by pressing the create event button. The event will now be created in the AWS database.

Table 2: Happy Path Workflow of uploading an event that reaches the database

Happy Path Workflow 2	**Initiative Tracker for DND use.**
Step 1	Select the tools button from the main menu screen.
Step 2	Select the DND section from the tools menu
Step 3	Once in the initiative tracker screen, the user will input the number of players.
Step 4	Upon receiving a number of players, the app will generate fields for that amount of players in which the user will enter names and initiative scores.
Step 5	User will then press the start button to sort the list of players by their initiative scores in descending order.
Step 6	Upon the completion of each players turn, a button is pressed to highlight the name of the player whose turn it is next.

Table 3: Happy Path Workflow of using the DND Initiative Tracker

Happy Path Workflow 3	**Bracket Generation.**
Step 1	Click on the Tools menu from the main menu screen.
Step 2	Within the Competitive events tab, click on the tournament creation button to begin bracket generation.
Step 3	Enter in the bracket parameters (double elimination, swiss, etc), number of players.
Step 4	User will be prompted to seed the players entered in the bracket. Once seeding is complete the user will press the generate bracket button, which will create the bracket.
Step 5	Selecting a match in the bracket will prompt the user to input the winner, and then will move the players in the bracket accordingly.

Table 4:Happy Path Workflow of create a tournament bracket

Unhappy Path Workflow 1	**Failed Log in.**
Step 1	Upon entering the app, user will be presented with a login screen.
Step 2	If the user input (email and password) do not have an associated account in the AWS database, an alert will display letting user know that the login info is invalid.
Step 3	Application will prompt the user to register an account or submit an email for a password reset.

Table 5: Unhappy Path Workflow of being unable to access Guild

Unhappy Path Workflow 2	**Connection to database fails**
Step 1	User searches for public events, published brackets, or uploads an event.
Step 2	Call to the database is made.
Step 3	Timeout occurs, users is alerted with a display which states that the connection to the database is faulty and the data cannot be retrieved.

Table 6: Unhappy Path Workflow when the database fails to connect

Unhappy Path Workflow 3	**Searching for a specific event type but no results are found**
Step 1	User navigates to the event manager screen by pressing the manage events button.
Step 2	Search public events button is pressed, taking user to a new screen.
Step 3	User enters in the parameters (Event type, game name, location, etc).
Step 4	Call to the database is made. Does not find a match for the given parameters. Instead, the application will loosen the search criteria. Then the user will be alerted that no events of that type exist, and return events that match the input as closely as possible.

Table 7: Unhappy Path Workflow if no events of a specified category is found

All of the happy path and unhappy path workflows will be validated upon the completion of their respective components with user testing. Users will follow the steps outlined and verify that the application handles the path in the way that is defined here.

Testing Strategy

Unit testing will be conducted every time a feature has been added or modified. This way the team is able to keep track of what individual components of the project are working and which may need to be modified. To perform unit testing, the team will use as many types of varied inputs (correct and incorrect) as possible and make sure all cases are handled correctly, alerting the user of failure and preventing crashes from invalid data. In addition to unit testing, the team has also been conducting user tests as frequently as possible, having advisors, stakeholders, and friends use the application on their own, and reporting back with bugs or changes that are necessary. Guild has even been used at a real gaming tournament. User testing will help us to ensure the application functions correctly as the users will not have the same information on the structure of the application and may wind up finding an issue the development team would not have found.

Those who will be responsible for testing Guild's story functionality will be Ryan Van, and the external advisors of Team #21. Since Ryan's primary responsibility is to develop the back-end of Guild, he does not work on developing the story with the team's front-end designers. The team's external advisors also do not develop the story for Guild as well since they play the role of mentors for the team and are only present to provide guidance. Ryan will be testing Guild's story at least once a week, or when a major change has been implemented into the app. The external advisors of Team #21 will test the app whenever the team has a meeting with them, or at least during the bi-weekly meetings the team has with the external advisors.

If a defect is found during testing, note of the defect will be taken. The defect will be reported by notifying all team members via instant messaging and documented in a separate text file. Actions to resolve the defect will take place after testing the whole application has been completed. To properly locate where the defect occurs, breakpoints will be placed in key areas of our program in Xamarin. An example of a defect the team could run into is pulling events from the database. Breakpoints would be placed before the button to retrieve the events is clicked, before and after the code calls the Flask URL, and after the section where data will be displayed to the user. If no issues were found during the use of breakpoints, then the team would look into the Flask code and MySQL database if an error is still present. This is to ensure that the proper names and syntax are being used in Xamarin, Flask, and the database for the tables and columns. If the issue is found to be related to UI, our front-end lead and front-end assistant, Riley Moore and Blake Cash, will conduct troubleshooting steps to determine what is causing the defect. If the issue is found to be related to the database, the back-end lead, Ryan Van, will research how the data is being translated or being

effected. The troubleshooting steps will mainly consist of looking through the internet and asking the team's external advisors for assistance.

Determining the 'completion' of an ambitious project like Guild is something the team has considered extensively. Given the team is only comprised of three members and Guild has an extremely extensive planned feature list, the odds of every single conceptualized feature being present on the application come May is very low. With this in mind, the team would like to stress the difference between Guild being 'finished' and Guild being in a state that the team considers worth showing off. The team will continue to actively add features and tools to Guild until time becomes a serious limiting factor, at which point the team will shift focus to refining and testing what is already in place in the application. In this way, the team does intend to continue development of Guild after Innovation Day, but when Innovation Day occurs Guild will nonetheless be a completely functional and useful application with an large feature list. As such, Guild's 'completion' state as far as this class is concerned is nebulous in it's exact feature list, but consistent in that UI design and database design will be finalized, the app will contain a generous (but not finalized) list of tools, and all bugs spotted through the testing process will eliminated.

The test plan for the acceptance criteria and workflow will consist having the testers use the application as if they have never used Guild before. The team will document how the testers go about using Guild. The plan will mainly consist of having the user create their Guild account, create an event, and join an existing event. We will ask the tester to run through all the tools provided on the app to ensure it's being displayed or functioning properly. The app will be ran on multiple devices to confirm if the team's UI functions properly on other devices as well since different devices have different screen sizes and resolution.

Team Contributions

Blake Cash: For this portion of the project, Blake Cash focused on the abstract, recent project changes, as well as working on drafting a testing strategy alongside the rest of the team. Like all other team members, Blake also helped proofread the assignment. In total, Blake worked for approximately two and a half hours.

Riley Moore: For Project Part #3, focused on the testing workflow, and laid out three happy path and unhappy path workflows of Guild. He also outlined the types of testing that would be conducted on all parts of the application. Along with the rest of the team, Riley proofread all parts of the project part. Riley spent approximately two hours on this assignment.

Ryan Van: For Project Part #3, Ryan Van focused on the User Stories and Acceptance Criteria section. In addition, he assisted in writing the Testing Strategy section for outlining how the team plans to handle defects and the test plan. Ryan also proofread the assignment with the other team members. Ryan spent around two hours working on this project.