Activitize

Sprint 2 Planning Document

Team 2:

Michael Golden, Richard Gorbett,

Tyler Preston, Tori Shurman

**Sprint Overview:** This sprint will focus on improving upon the stories implemented in sprint one, and adding new features to make the application more social media oriented. Team members will diversify their tasks, as they attempt to become skilled in developing on the frontend and the backend. Communication between the frontend and backend will also be improved in this sprint so that more data may populate for the users to see.

**Scrum Master:** Tori Shurman

**Meeting Schedule**: Tue/Thur 3-4:00

**Risks/Challenges:** The first sprint was largely setup, and was therefore challenging when it came to learning new frameworks and technologies. As this sprint focuses on implementing more logically complex features, we expect this sprint to be more mentally challenging as it will require application of complex logic in order to build a robust application.

User Story 1: As a developer, I would like the user to be able to review the app, providing questions, comments, and concerns.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Add a button on the frontend to link to a feedback form, | 3 hours | Rich |
| 2. | Set up an email account for the app to receive feedback emails | 2 hour | Michael |
| 3. | Create a feedback form on the frontend, which emails us with user feedback | 6 hours | Michael |

Acceptance Criteria:

* Have a clickable button for feedback
* Have feedback be sent to a feedback email

User Story 2: As a user, I would like to modify events I have created.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Improve upon user story 4 from sprint 1, providing the ability to view an event | 5 hours | Tori |
| 2. | In the user’s profile, provide a button to link to all of their created events | 4 hours | Rich |
| 3. | When the user taps one of his or her created events, allow for an edit button | 3 hours | Rich |
| 4. | Create an edit event view on the frontend, similar to the create event view | 4 hours | Tori |
| 5. | Implement ability to store and modify event data on the backend | 4 hours | Tori |

Acceptance Criteria:

* Be able to change event information
* Be able to locate events created by user
* Update back end with updated event

User Story 3: As a user, I would like to rsvp to events I am invited to.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Task Description |  | Estimated Time | Owner |
|  | Implement an option to invite people in the front-end |  | 4 hours | Rich |
| 2. | Be able to state that you are either going or not going to an event |  | 4 hours | Rich |
| 3. | Be able to view who is going to an event |  | 4 hours | Tyler |
| 4. | Be able to view who is not going to an event |  | 4 hours | Tyler |

Acceptance Criteria:

* The user should be able to send invites to people
* The invited people should be able to mark whether they are going or not going to an event
* The server should save the people that are going and not going to the event
* Users should be able to view who is going and not going to an event

User Story 4: As a developer, I would like to allow users only 3 failed login attempts.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Improve user story 1 from sprint 1, improving login so it is more secure and communicates more with the backend | 4 hours | Tori |
| 2. | Allow handling of incorrect login attempts on the front end, capping the limit at 3 | 5 hours | Michael |
| 3. | Implement an appropriate error response when incorrect login credentials are received on the backend | 4 hours | Rich |
| 4. | Prevent further login attempts for a set amount of time | 4 hours | Michael |

Acceptance Criteria:

* After 3 failed login attempts the app, lock account for increasing periods of time
* If an incorrect user name or password is received notify the user of the app
* Prevent login attempts even if the user closes the app and restarts it, if the account is locked.

User Story 5: As a user, I would like to favorite events I have created or am invited to.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Improve upon user story 4 from sprint 1, modifying the event view and the event feed to have a “favorite” button | 4 hours | Rich |
| 2. | Implement the ability to add a favorite feature on the backend | 4 hours | Rich |
| 3. | Implement a view that has favorited events | 6 hours | Michael |

Acceptance Criteria:

* Be able to favorite events
* Be able to see favorited events

User Story 6: As a user, I would like to be able to create groups of people that I can participate in events with.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Improve user story 2 from sprint 1, adding a friends and groups feature to the profile | 4 hours | Michael |
| 2. | Implement the ability to handle a user’s groups on the backend | 5 hours | Tori |
| 3. | When a user taps on groups, allow them to add a new group | 5 hours | Michael |
| 4. | When a user taps on friends and groups, allow them to view existing friends and groups | 5 hours | Michael |
| 5. | Modify user story 3 from sprint 1, allowing the “Invite” field of edit event to populate with the user’s groups | 6 hours | Michael |

Acceptance Criteria:

* Be able to create groups of people
* Be able to invite a whole group of people to an event

User Story 7: As a user, I would like to be able to remove people from events I have created.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Implement the ability to modify an event on the frontend | 5 hours | Tori |
| 2. | Implement the ability to remove a user from an event on the frontend | 5 hours | Rich |
| 3. | Implement the ability to remove a user from an event on the backend | 5 hours | Tyler |
| 4. | When a user is removed from an event, remove that event from the events they are invited | 5 hours | Rich |

Acceptance Criteria:

* Be able to edit an event
* Be able to remove people from an event
* Once a person is removed from an event, that event should not appear in their list of events

User Story 8: As a developer, I would like to incorporate a three-tier architecture.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Ensure functionality and finalize setup of backend | 10 hours | Tyler |
| 2. | Develop base functionality in all three tiers | 10 hours | Tyler |

Acceptance Criteria:

* Proper functionality is achieved given the appropriate architecture.

User Story 9: As a developer, I would like to separate business logic from client logic.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task Description | Estimated Time | Owner |
| 1. | Completely setup MVC framework | 10 hours | Tori |
| 2. | Add in base functionality and placeholders to ease in future development | 10 hours | Tyler |

Acceptance Criteria:

* No business logic and client logic is intertwined

Remaining Product Backlog:

Functional

1. As a user, I would like to create sub-events, i.e. events that are created for people attending the main event.
2. As a user, I would like to be able to mute notifications from events I am not interested in.
3. As a user, I would like to have profile that can be viewed by others.
4. As a user, I would like to add friends.
5. As a user, I would like to search for new friends.
6. As a user, I would like to extend invites to friends for existing events.
7. As a user, I would like not to be required to change my password often.
8. As a developer, I would like there to be minimal time to load information on events and events of friends for the user.
9. As a user, I would like to not have ads.
10. As a user, I would like to know that all of my information is secure.
11. As a user, I would like to be able to comment on events that I have created or am invited to.

Non-Functional

1. Should be able to utilize Google services in some-form
2. Should use some sort of near-real-time or real-time solution for updates
3. Should be scalable to be user demand
4. Should meet standard security practices
5. Should be able to have negligible downtime while allowing for continuous development
6. Should be able to be high performing while maintaining proper security practices
7. Should document and provide API for other services to be able to hook into