Sprint 1 Planning Document

Team 3 - HiLingual

Garrett Davidson, Noah Maxey, Nate Ohlson, Joseph Savastano, Riley Shaw, Vincent Zhang

Table of Contents

1. Sprint Overview 3
    1.1. Scrum Master 3
    1.2. Meeting Schedule 3
    1.3. Challenges 3
2. Current Sprint Detail
    2.1. User Story 1 4
    2.2. User Story 2 5
    2.3. User Story 3 6
    2.4. User Story 4 7
    2.5. User Story 5 8
    2.6. User Story 6 9
3. Remaining Backlog 10

Sprint Overview

Our first sprint is going to focus on basic functionality of our application. By the end of the sprint the user will be able to create an account, login, manage their profile, and navigate through the majority of our GUI. The overarching structure of the application will be instantiated.

SCRUM Master: Garrett Davidson

Meeting scheduled: Monday, Wednesday, and Friday at 12:20, Tuesday and Thursday at 1:30

Challenges: Most team members are not familiar with Xcode and Swift. In addition, most team members are not familiar with server side technologies and frameworks, such as Dropwizard, MySQL, and Redis. This first sprint will require a considerable amount of time to learn and become familiar with these new languages and frameworks. This is also the first time the team members have worked together and so we must ensure that we are working together efficiently and effectively.

Current Sprint Detail

User Story #1

As a user, I would like to set up an account via Facebook or Google

#	Task Description	Estimated Time (Hours)	Owner
1.	Create iOS project skeleton	6	Garrett
2.	Create server project skeleton	5	Vince
3.	Register domain	2	Nate
4.	Set up production server	4	Nate
5.	Implement client-side Facebook authentication	6	Riley
6.	Implement client-side Google authentication	6	Joey
7.	Implement server-side authentication	4	Vince
8.	Set up databases with API server	4	Nate
9.	Implement “create account” user interface	3	Noah
10.	Implement data models (server)	5	Joey
11.	Implement data model persistence (server)	6	Joey
12.	Create profile page UI	4	Noah
13.	Allow user to modify profile	4	Riley
14.	Create and update user in database with profile info	4	Joey
Total Estimated Time		63	

Acceptance Criteria:

- Given that a user can successfully log in, when they want to change their information, then they should be prompted to initialize their profile, and those changes will be reflected in the database.
- Given that a user has successfully logged in and built a basic profile, when they use the app, they should be able to access the full functionality of HiLingual.

User Story #2

As a user, I would like to log into an account via Facebook or Google.

#	Task Description	Estimated Time (Hours)	Owner
1.	Implement client-side Facebook/Google authentication	5	Noah
2.	Implement client interface for logging in via Facebook/Google	3	Noah
3.	Implement server-side authentication verification	5	Vince
4.	Implement database pulling Facebook/Google information	5	Garrett
5.	Retrieve user data from the database	3	Vince
Total Estimated Time		21	

Acceptance Criteria

- Given that a user has successfully gone through the account creation process, when they decide to log in, then they should be able to log in to their account.
- Given that the login process has been implemented properly, when a user logs in, the app should load all of the user’s information and settings.

User Story #3

As a user, I would like to choose and converse with users who speak the language I am trying to learn.

#	Task Description	Estimated Time (Hours)	Owner
1.	Create UI for matching with users	4	Garrett
2.	Find suitable matches for the search	6	Riley
3.	Create UI for browsing potential matches	6	Garrett
4.	Implement starting conversations with matches	4	Riley
6.	Implement accepting conversation requests	4	Noah
7.	Implement conversation list	5	Garrett
Total Estimated Time		29	

Acceptance Criteria:

- Given that a user has successfully created an account, when the user navigates to the find users tab, they should be able to look through a list of recommended users and be able to choose other users to chat with. They will only be able to add them to their conversations and not send messages.
- Given that the user is able to go through a list of recommended users, when they look at their conversation list, they should be able to see the users that they have added in order to be able to chat with them in a later sprint.
- Given that the user is able to see the list of users that they have added, when a user selects another user, they should be able to view that user's profile including all of the information they have added to their profile.

User Story #4

As a user, I would like to search for specific users by name.

#	Task Description	Estimated Time (Hours)	Owner
1.	Create Search UI	6	Noah
2.	Implement search based on name	8	Nate
3.	Implement search based on username	6	Nate
Total Estimated Time		20	

Acceptance Criteria:

- Given that search has been implemented properly, when a user searches for another user, if a user exists within the database then another user should be able to find them.
- Given that a user accepts another user to their conversation list, when they go to their conversation list, then they should be able to find that person without searching for them again.

User Story #5

As a user, I would like to edit my personal information on my profile.

#	Task Description	Estimated Time (Hours)	Owner
1.	Create UI for user to access profile	4	Nate
2.	Implement changing profile settings	4	Joey
3.	Implement changing profile information	4	Riley
4.	Implement pictures from Facebook/Google	3	Garrett
5.	Implement photo uploading/taking	5	Riley
Total Estimated Time		20	

Acceptance Criteria:

- Given that we have properly implemented Google/Facebook integration, when the user links their account, information and pictures should be pulled from the respective accounts.
- Given that we have properly implemented the user profile, when a user makes changes to their profile, the changes should be reflected on the server.

User Story #6

As a user, I would like to receive push notifications.

#	Task Description	Estimated Time (Hours)	Owner
1.	Register and integrate app with APNs on device	5	Joey
2.	Register service with APNs (get certs, etc)	5	Nate
3.	Implement APNs provider (server)	10	Vince
4.	Configure APNs provider with APNs	5	Noah
5.	Send push notifications via provider	2	Vince
Total Estimated Time		27	

Acceptance Criteria:

- Given that the APNs provider has been implemented and configured properly and the user’s device is reachable, when the API server sends a push notification, the user will receive it.
- Given the user has properly received a push notification, when they receive it, they should be able to select the notification and be taken to the portion of the app that the notification is referring to.

Remaining Backlog

Functional:

1. As a user, I would like to send messages to other users in real time.
2. As a user, I would like to correct messages sent by other users.
3. As a user, I would like to be able to translate messages that a user sent me.
4. As a user, I would like to send and receive voice messages.
5. As a user, I would like to create flashcards for vocabulary words.
6. As a user, I would like to review flashcards.
7. As a user, I would like to maintain multiple flash card sets.
8. As a user, I would like to be able to report spam.
9. As a user, I would like to block specific users.
10. As a user, I would like to be able to pay to increase my daily match limit.
11. As a user, I would like to be able to select my interests to find better matches (if time allows).
12. As a user, I would like to match with people with whom I share similar hobbies or interests (if time allows).
13. As a user, I would like to be able to pay to increase my daily message translation limit (if time allows).
14. As a user, I would like to be able to pay to increase my daily match limit (if time allows).
15. As a developer, I would like to see insights regarding how much interaction users have with each of their matches, to be able to improve the algorithm (if time allows).

Non-Functional:

1. Interface should be easy to use and intuitive to navigate.
2. Messaging must be as fast and reliable as possible.
3. App-server communications must be done securely.
4. App must be resilient to network connectivity disruptions.
5. Server must be able to integrate with various support APIs (translation, etc).
6. Server must support a large number of concurrent users.
7. Platform must support automatic staging and deployment (if times allows).
8. Server must provide a maintenance/administration panel (if time allows).
9. App must have an algorithm to match users based on language abilities and shared interests (if time allows).