

Voice Capsule Sprint Scrum Board History

Key:

USX - user story #X

USX.Y - task #Y as a part of user story #X

Sprint 1

Initial Configuration

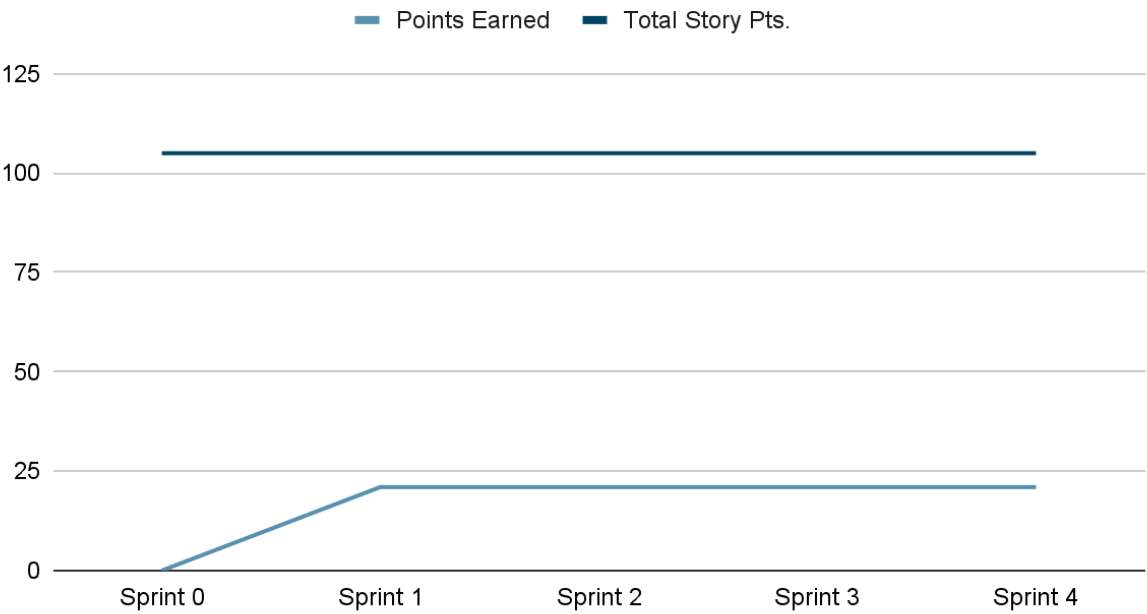
To-Do	In-Progress	Done
US1: As someone who values privacy, I want to be able to use an app which lets me log in and out of an account securely		
US1.1: Research an app framework to use		
US1.2: Gain familiarity with Flutter and Dart		
US1.3: Make a GitHub account and upload the release and sprint plans		
US1.4: Update the sprint plan with specific tasks about logging in after framework research		
US1.5: Begin designing the user account and authentication system		
US2: As a database engineer, I value systems that handle data safely		
US2.1: Researching which database to use		
US2.2: Getting familiar with how to use the database		
US2.3: Update this Sprint plan with more specific tasks		

about the login once research on database has been done		
US2.4: Once database system has been agreed upon, begin integrating the database into the app		

Final Configuration

To-Do	In-Progress	Done
		US1: As someone who values privacy, I want to be able to use an app which lets me log in and out of an account securely
		US1.1: Research an app framework to use
		US1.2: Gain familiarity with Flutter and Dart
		US1.3: Make a GitHub account and upload the release and sprint plans
		US1.4: Update the sprint plan with specific tasks about logging in after framework research
		US1.5: Begin designing the user account and authentication system
		US2: As a database engineer, I value systems that handle data safely
		US2.1: Researching which database to use
		US2.2: Getting familiar with how to use the database
		US2.3: Update this Sprint plan with more specific tasks about the login once research on database has been done
		US2.4: Once database system has been agreed upon, begin integrating the database into the app

Burnup Chart



Sprint 2

Initial Configuration

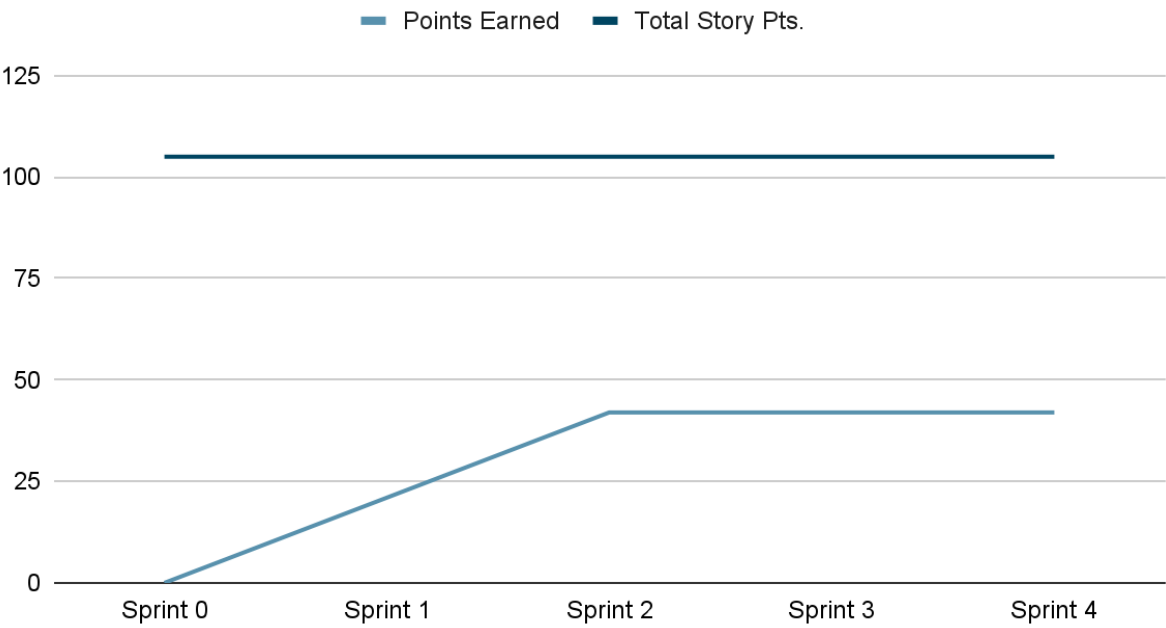
[illegible]

Final Configuration

To-Do	In-Progress	Done
	US2: As someone who loves hearing messages from the past, I want to be able to download and save voice notes that have been sent to me	US1: As a mindful individual, I want to send my future-self positive messages and affirmations
	US2.3: Download/upload data from/to Firebase	US1.1: Add a button
	US2.4: List all of the recorded voice notes and allow for playback (only after an indicated time)	US1.2: Introduce audio recording functionality
		US1.3: Attach button to mic recording function
		US1.4: Include a time elapsed indicator and timer for the voice clip playback
		US1.5: Link button to time elapsed display for a voice clip
		US1.6: Introduce a limit on the length of a voice clip recorded
		US1.7: (Spike) Learn how to use the microphone and necessary APIs
		US2.1: Introduce a button that gives the option to send a voice clip to yourself
		US2.2: Add a page after recording that allows a user to select one's self (or eventually a contact) as the recipient of a voice clip
		US2.5: (Spike) Discuss protocol for determining when

		a user can fetch a voice clip
--	--	-------------------------------

Burnup Chart



Sprint 3

Initial Configuration

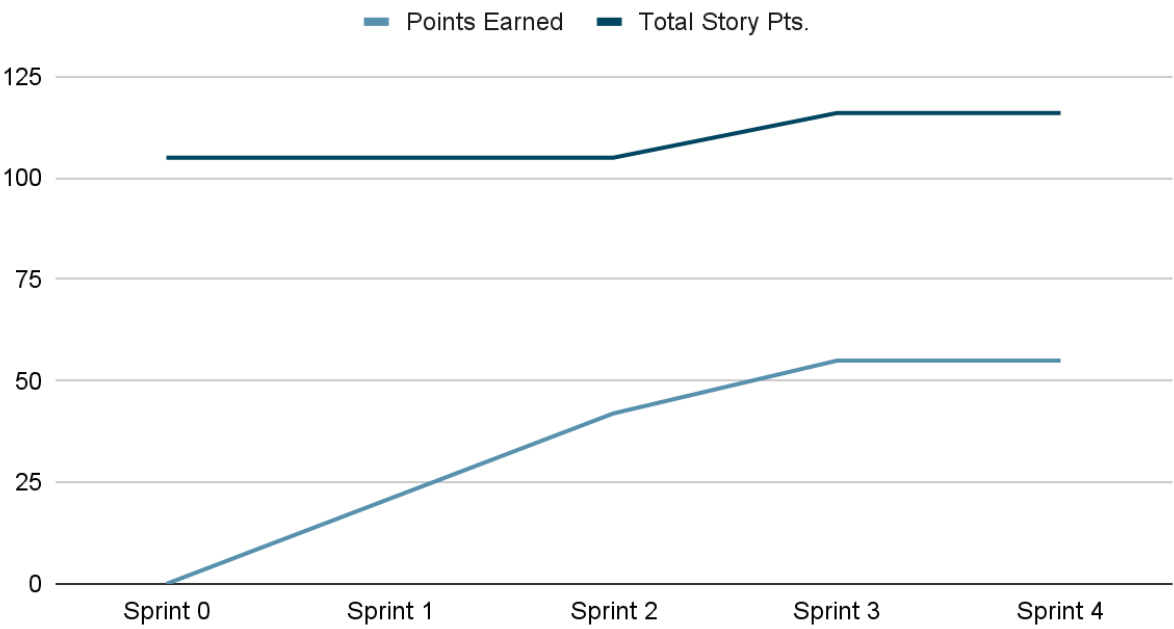
To-Do	In-Progress	Done
US1: As someone who loves hearing messages from the past, I want to be able to download and save voice notes that have been sent to me		
US1.1: Send functionality connected to Firebase - allowing users to send voice notes to themselves		
US1.2: Upon sending, the user should be able to decide if the voice note can be downloaded and saved by the recipient, or whether it will automatically perish after having been listened to once.		
US1.3: User should be able to download opened voice notes		
US1.4: Users should be able to delete all voice notes that they have received		
US2: As a sociable person, I want to be able to search for and add other users as "friends" on the app easily		
US2.1: Create 'add friends' page		
US2.2: Add search bar		
US2.3: Make search bar functional - will be able to search for other users based on their username		
US2.4: Allow users to accept		

or decline friend invites		
US2.5: Have a page that lists all friends		
US3: As a forgetful person, I want to send my friend a happy birthday message eight months in advance, but they can only open it on their birthday		
US3.1: The voice notes should have option to be opened on specified date - look into calendar library		
US3.2: Connect this info to firebase		
US3.3: Ensure that the voice note can only be opened if it is the specified day		

Final Configuration

To-Do	In-Progress	Done
US2.2: Add search bar (<i>feature dropped</i>)	US2: As a sociable person, I want to be able to search for and add other users as “friends” on the app easily	US1: As someone who loves hearing messages from the past, I want to be able to download and save voice notes that have been sent to me
US2.3: Make search bar functional - will be able to search for other users based on their username (<i>feature dropped</i>)	US2.4: Allow users to accept or decline friend invites	US1.1: Send functionality connected to Firebase - allowing users to send voice notes to themselves
		US1.2: Upon sending, the user should be able to decide if the voice note can be downloaded and saved by the recipient, or whether it will automatically perish after having been listened to once.
		US1.3: User should be able to download opened voice notes
		US1.4: Users should be able to delete all voice notes that they have received
		US2.1: Create ‘add friends’ page
		US2.5: Have a page that lists all friends

Burnup Chart



Sprint 4

Initial Configuration

To-Do	In-Progress	Done
US1: As a sociable person, I want to be able to search for and add other users as friends on the app easily.		
US1.1: Add friends by user email		
US1.2: Create a page that lists a user's contacts		
US1.3: Adding a friend updates the current user's contacts in Firestore database		
US1.4: Users should receive a notification that they have pending friend invites		
US2: As a forgetful person, I want to send my friend a happy birthday message eight months in advance, but they can only open it on their birthday.		
US2.1: Use calendar library to be able to choose open date/time		
US2.2: Create voice capsule data structure with sender, receiver, open date/time, and database info fields		
US2.3: Ensure that voice capsules can only be opened after the specified open date/time		
US3: As a technophobe, I want to be able to use an app		

that is simple to navigate and use		
US3.1: Change button styles		
US3.2: Change logout button to navigation bar		
US3.3: Formulate style/color scheme for the app		
US4: As someone who loves hearing messages from the past, I want to be able to download and save voice notes that have been sent to me		
US4.1: Voice Capsules are deleted from the database once they have been opened		
US4.2: Receiver can download opened voice notes to local storage		
US4.3: Allow users to accept or decline friend invites		
US4.4: Design Firestore database collection/document structure		
US4.5: Upload file to Firebase Storage and update Firestore Database		
US4.6: Query Firestore database for openable voice capsules and download them from Firebase Storage		

Final Configuration

To-Do	In-Progress	Done
		US1: As a sociable person, I want to be able to search for and add other users as friends on the app easily.
		US1.1: Add friends by user email
		US1.2: Create a page that lists a user's contacts
		US1.3: Adding a friend updates the current user's contacts in Firestore database
		US1.4: Users should receive a notification that they have pending friend invites
		US2: As a forgetful person, I want to send my friend a happy birthday message eight months in advance, but they can only open it on their birthday.
		US2.1: Use calendar library to be able to choose open date/time
		US2.2: Create voice capsule data structure with sender, receiver, open date/time, and database info fields
		US2.3: Ensure that voice capsules can only be opened after the specified open date/time
		US3: As a technophobe, I want to be able to use an app that is simple to navigate and

		use
		US3.1: Change button styles
		US3.2: Change logout button to navigation bar
		US3.3: Formulate style/color scheme for the app
		US4: As someone who loves hearing messages from the past, I want to be able to download and save voice notes that have been sent to me
		US4.1: Voice Capsules are deleted from the database once they have been opened
		US4.2: Receiver can download opened voice notes to local storage
		US4.3: Allow users to accept or decline friend invites
		US4.4: Design Firestore database collection/document structure
		US4.5: Upload file to Firebase Storage and update Firestore Database
		US4.6: Query Firestore database for openable voice capsules and download them from Firebase Storage

Burnup Chart

