AntennaPod - Team Omega
Release Summary 2

Team members

Name and Student ID	GitHub id	Number of story points that member was an __author__ on.	
		Iteration 3	**Iteration 4**
Raphaëlle Giraud	Raph1105		- Discovery tab (8 points) - Preferences (5 points)
Sarbeng Frimpong	jiggy357		- Add podcast to folders (5/8 points)
François Crispo-Sauvé	FrankSauve	- Sub-categories on the window (≈4 points)	- Autocomplete suggestions (4 points)
William Kingbede	williamkingbede	- Advanced search page (13 points)	- Create folders (13 points)
William Leclerc	LeCleric	- Sub-categories on the window (≈4 points)	- Autocomplete suggestions (4 points)
Constandinos Papadakos	Cotso		- Folders tab (4/8 points)

During the third iteration we were busy with midterms and other work, therefore only three of us were able to complete some stories. For iteration four, since all the midterms had been completed by then, the team was able to work at full strength again in order to implement new stories as well as stories that were incomplete from iteration 3.

Velocity

For this part, please refer to the "Planning poker and burndown chart R2" document, situated in the CourseAdmin file.

	TOTAL: 8 stories, 68 points over 5 weeks		
	Number of stories	**Number of points**	**Milestone link**
Iteration 3	2	21	Milestone 3
Iteration 4	6	50	Milestone 4

Overall Arch and Class diagram

Link to all the UML class diagrams

The only changes are the addition of a couple fragments and adapters without modifying the overall architecture. Any changes are listed on the story issues.

Plan up to next release

For this part, please refer to the "Planning poker and burndown chart R2" document, situated in the CourseAdmin file.

	TOTAL: 11 stories, 95 points over 4 weeks		
	Number of stories	**Number of points**	**Milestone link**
Iteration 5	5	45	Milestone 5
Iteration 6	6	50	Milestone 6

Infrastructure

- Constraint Layout: This library allows us to create large and complex layouts by simplifying the layout editor.

Name Conventions

- Oracle Java naming conventions for our class names, methods names and variable names.

- [Ribot Android naming conventions](#) for our layout files, drawable files and general code guidelines.

Code

File path with GitHub link	Purpose
[390AntennaPod/app/src/main/java/de/danoeh/antennapod/fragment/ItunesSearchFragment.java](#)	This is where advanced search, subcategories and autocomplete was implemented.
[390AntennaPod/app/src/main/java/de/danoeh/antennapod/adapter/DiscoveryAdapter.java](#)	Contains all the action listeners for the discovery tab.
[390AntennaPod/app/src/main/java/de/danoeh/antennapod/fragment/DiscoveryFragment.java](#)	Contains the methods to load the podcasts based on the preferences of the user.
[390AntennaPod/app/src/main/java/de/danoeh/antennapod/fragment/FoldersFragment.java](#)	Contains the implementation of the folders window using a similar layout to the subscription window.
[390AntennaPod/app/src/main/java/de/danoeh/antennapod/dialog/NameFolderDialog.java](#)	The class is used to create a popup dialog to name the newly created folder.

Testing

- [Unit tests:](#)

Test File path with GitHub link	What it is testing
https://github.com/FrankSauve/390AntennaPod/blob/develop/app/src/androidTest/java/de/test/antennapod/itunes/ItunesApiTest.java	Testing Itunes fragment search and autocomplete functions. One test (testItunesArtistSearch()) has been commented out because of failure in CI build even though the test passes on local machines.
https://github.com/FrankSauve/390AntennaPod/blob/develop/app/src/androidTest/java/de/test/antennapod/folders/FoldersTest.java	Testing the creation of a folder behaviour. As the database has not been mocked, folders can be deleted from database at the end of the test.
https://github.com/FrankSauve/390AntennaPod/blob/develop/app/src/androidTest/java/de/test/antennapod/discovery/DiscoveryTest.java	Testing discovery fragment main functionality.

- Ui tests:

Test File path with GitHub link	What it is testing
https://github.com/FrankSauve/390AntennaPod/blob/develop/app/src/androidTest/java/de/test/antennapod/ui/MainActivityTest.java	Testing the MainActivity user interface functionalities. Most of the new stories UI tests are added in this test file as they most of the time reference to the MainActivity.
https://github.com/FrankSauve/390AntennaPod/blob/develop/app/src/androidTest/java/de/test/antennapod/ui/PreferencesTest.java	Test the PreferenceActivity user interface functionalities. UI test for discovery preferences function has been added in this file.

Finished SHORT Story summaries

Points: 8, Priority: High, Risk: Low, Iteration 3
Story - Sub-categories on the window: As a user, I want to have a sub-categories displayed on the window so I can find new podcasts of a specific genre.
Feature: Sub-Categories on the window

       This story involved allowing a user to find podcasts based off of what category/genre they were a part of. Since we were using the Itunes search, we used their category system's IDs. Kingbede provided a base for Frank and Leclerc to implement the feature with by completing his Advanced search feature. Leclerc and Frank worked on Unit testing and Leclerc also implemented UI tests once the feature had been completed.

       An example of the feature in action is if a user wished to look for comedy podcasts they could browse podcasts of only that category, ignoring other types.

Points: 13, Priority: Medium, Risk: Medium, Iteration 3
Story - Advanced search page: As a user, I want to be able to, in the search page, seek for podcasts by category, language, artist (and date) so that I can easily find what I am looking for specifically.
Feature: Advanced search

       This story involved modifying the iTunes search to include artist, title, language and category search. Kingbede and Leclerc both worked on unit and UI tests.

Points: 5, Priority: Medium, Risk: Low, Iteration 4
Story - Autocomplete suggestions: As a user, I want to have suggestions in the search bar, so I can be more productive in my research.
Feature: Autocomplete for itunes search

       This story involved enhancing the iTunes podcast search to provide suggestions for the user based on what they were currently typing. This was implemented and unit tested by Frank, with Leclerc also providing some tests as well.

Points: 8, Priority: Medium, Risk: Medium, [Iteration 4](#)
[Story - Folders tab](#): As a user, I want to be able to see my folders in the menu, so that I can access them easily.
Feature: [Create folders](#)

This story involved allowing a user to quickly view all their created folders from the Navigation Drawer. Costa is currently working on fully implementing and testing this story.

Points: 8, Priority: Medium, Risk: Medium, [Iteration 4](#)
[Story - Add podcasts to folders](#): As a user, I want to be able to add podcasts (from the episodes page) into the folders that I have created, so that I can populate them.
Feature: [Create folders](#)

This story involved being able to add podcasts to the user-generated folders. Currently, there is code that should allow for the addition, but it is not fully implemented. Sarb was the one who worked on this story.

Points: 13, Priority: Medium, Risk: High, [Iteration 4](#)
[Story - Create folders](#): As a user, I want to be able to create folders (and name them), so that I organize my podcasts.
Feature: [Create folders](#)

This story involved allowing users to create folders and name them. Kingbede fully implemented this story and provided unit tests alongside ui tests.

Points: 8, Priority: Medium, Risk: Medium, [Iteration 4](#)
[Story - Discovery tab](#): As a user I want to be able to have a Discovery page so that I can have recommendations according to my preferences (entered in my preferences profile).
Feature: [Discovery page](#)

This story involved generating a list of podcasts that the user may be interested in based on their pre-defined preferences. This story ties into the preferences story. Therefore there would need to be a discovery tab in the menu and a page that the tab leads to which displays the suggestions. Raphaelle worked on this story, wrote the code, and provided unit and UI tests for it.

Points: 5, Priority: Medium, Risk: Medium, [Iteration 4](#)
[Story - Preferences](#): As a user I want to be able to have my preferences stored so that I can have a preferences profile in the app.
Feature: [Discovery page](#)

This story involved having the user outline their preferences for podcasts on the settings page in order to generate the Discovery page's content. Raphaelle worked on this story, wrote the code, and provided a UI test for it. Its functionality was also tested in a unit test for the discovery tab story.

CI stages and notification

CI Stages

The stages of continuous integration are described by the [circle.yml](#) file:

- Machine: This is where the machine that runs the build is setup. It downloads or updates the android SDKs that are necessary for AntennaPod and it sets java to run on oraclejdk8.

- **Build & Test:** The Android emulator is booted with API 24 and is ran in the background. When the emulator is successfully booted, the build starts with the command *./gradlew assembleDebug* and then the unit tests run on the build using *./gradlew connectedFreeDebugAndroidTest.* If any of these stages fail, circleCI will report a failing build and stop the execution.

- **Post:** This is where the html test reports are collected and copied in the artifacts folder so it can be viewed on the circleCI website.

CI Notifications

Our entire team is notified in Slack when a build succeeds or fails with links to the build and the commit. This also allows the team members to discuss the builds in the same channel. Here are a few examples: