CMPE 451.01 - Milestone 2 Report

Looking for Concerts

Group 2

December 2, 2017

Contents

1 Executive Summary

1.1 Description

Looking for Concerts is a platform for musical concert-goers, where people from all ages come together to experience the warmth of a live concert stage with thousands of other music lovers. After a user signs up, he can create concerts, follow or be followed by friends, search for concerts, see his friends' activity and attend concerts! Our intelligent recommendation engine will quickly pick up his interests. Especially if he connects his LFC account with Spotify. What's more, he can find past concerts and get lost in the comments, upvoting, downvoting and reacting! We would like to provide an immersive experience and help concert-goers find both exciting concerts to go to and interesting people to go to those concerts with.

1.2 Technical overview

1.2.1 Back-end

Our Looking for Concerts (LFC) API is powered by the Django framework. Kemal Berk, Haluk and Enes were in charge of creating the class models based on the class diagram on our Wiki page and implementing the endpoints. There were minor modifications in classes. For example, we do not have a class called "Concert Page" in our API, since it will actually be a part of the front-end. The endpoints implemented in the second milestone are: deactivate account, general and advanced search, follow/ unfollow users, attend/unattend concerts, rate concerts, connect/disconnect Spotify, upload image to concert pages and user profiles, get recommendations depending on liked/watched artists. We also changed the Rating model, now offering four different domains for ratings and reconstructed the Artist model according to the data we get from Spotify artists.

1.2.2 Front-end

In front end Ozan, Burak, Oğuzhan and Alpertunga were managing the user interactions for the Looking for Concerts. On the milestone 1 we were using Angular 2 but it was difficult to learn and we had hard times using it so we switched to use React to create our web application. In the end of this milestone we now have concert detail page, profile page, login/signup page and homepage. In concert page we added rate concert feature, we implemented attend concert functionality, we added concert tags, artist image and buy ticket button. In profile page there is follow/unfollow button for users to follow each other. We also show user data, user image, followers, following users and concert history and future concerts. There's a top navigation component implemented which has search functionality and some links to our components. Redux is implemented to keep track of tokens and user infos in the main store which keep the app reliable and easy to improve.

1.2.3 Android (mobile)

For Android, we used Java to develop our application. Elif, Furkan and Pınar were responsible for developing the app. Right now, with our application a registered user can login, create a concert, attend/unattend to a concert, make detailed rating and comment on a concert. The design of our application is a little better when we compare with previous milestone but we plan to improve it more so that it is user friendly and easy to use. User can know the exact location (venue and coordinates) of a concert while s/he is creating it. We have used Google Places API for this purpose. User can make artist search while creating concert by connecting with Spotify API. Also, user can make a basic search. For next milestone, we will give the user a chance to connect their account to Spotify. We will also add semantic tags. Right now, users just enter tags, but they don't mean anything. We will search for those tags and the user will be able to select the tag s/he wants to add. Also, user will login with Spotify and we will give concert recommendations to user via his/her Spotify account.

2 Deliverables Status List

FUNCTIONALITIES	Back-end	Front-end	Android
Concert Creation			
Search Artists from Spotify	✓	✗	✓
Search Tags from Semantic Web	✓	✗	✗
Add Image To Concert	✓	✗	✗
Add Google Maps Location	N/A	✗	✓
Add Seller Url	✓	✗	✓
Concert Interaction			
Attend / Cancel Concert	✓	✓	✓
Rate Concert	✓	✓	✓
Comment Concert	✓	✓	✓
Show Average Ratings	✓	✓	✓
Show Seller Url	✓	✓	✗
Show Concert Location Map	N/A	✓	✗
User Creation			
Add Image to User	✓	✗	✗
Authentication & Authorization			
JWT Authentication	✓	✓	✓
User Interaction			
Connect Spotify Account	✓	✗	✗
Get / Show Spotify Info	✓	✗	✗
Login with Spotify	✗	✗	✗
Follow/Unfollow User	✓	✓	✗
Search			
Basic Search	✓	✗	✓
Advanced Search	✓	✗	✗
Recommendation			
Get / Show Recommendations	✓	✗	✗
React Integration	N/A	✓	N/A
Deployment	✓	✗	✓

Table 1: Deliverables

3 Evaluation of Deliverables & Their Impact on Progress

1. **Concert Creation**

   (a) Search Artists from Spotify: With this we were able to use a vast knowledge base on artists and their images. Using Spotify artists as artists for concerts shall make recommendations easier in upcoming improvements since related artists info could be accessed from Spotify. Our customer specifically asked us to implement this feature, so it was important and affected our progress.

   (b) Add Image To Concert: Even though it gets artists image by default, for events that having an informative poster of its own, this functionality is a must.

   (c) Add Google Maps Location: With this, creator of the concert could give exact location of the concert. On the concert page this provides a great convenience to the concert goers as they can see where is the concert from google maps.

   (d) Add Seller Url: We provide a url of the page where the concerts are sold. We believe that this increases the comfort of the user, since he/she is able to contact with the seller with only one click.

2. **Concert Interaction**

   (a) Attend/Cancel Concert: This feature is very important as it is a preliminary for other functions. For example, users can only rate concerts to which they attended or will attend. We fully implemented the functionality.

   (b) Rate Concert: Ratings provide a crowd-sourced method of getting a signal for how good a concert is and therefore important. Furthermore, only one dimension is not sufficient, because someone can like the music but dislike the concert atmosphere. So, we separated ratings into 4 categories. Implementation in backend and frontend is complete but not in Android yet.

   (c) Comment on Concert: Commenting on concerts give in depth information about whether other users liked a concert. We were asked to also have categories for comments so that users can use it for filtering purposes. Users can currently comment and see comments. We will add the categories and comment filtering for the next milestone.

   (d) Show Average Ratings: Showing average ratings was not in our plan but we implemented it as an extra functionality. Android does not have it yet and frontend needs to improve the way the information is displayed.

   (e) Show Seller URL: Showing the URL where the users can buy the tickets is not critical and would not impact our plan but it was a nice addition. For the third milestone, we should check if the provided URL is valid or perhaps the reporting system will take care of it.

   (f) Show Concert Location Map: This feature is important for the usability of our application. The users can look at the to have a better sense of the address. Because our application is about finding and attending concerts, helping the users find the concert address is a must.

3. **User Creation**

   (a) Add Image to User: Although optional, adding an image to the profile page is widely practiced in all social media and it enhances the user experience a lot. Therefore, we decided to prioritize it and not wait for the third milestone. It is working but might need some improvements. It did not impact our progress.

4. **Authentication & Authorization**

   (a) JWT Authentication: After milestone 1, we had decided to switch to Json Web Token (JWT) Authentication as it provided two types of tokens: refresh and access along with expiration times. It was required for security and identifying the users and on the critical path in our plan since many features depend on authentication. We have it in all three teams: backend, frontend and Android.

5. **User Interaction**

   (a) Connect/disconnect Spotify Account: The main reason for this feature is that we want to use Spotify data for recommendations to overcome the ”cold start” problem. For now, we have it in our backend only, but we will integrate with frontend and Android soon. We also offer disconnecting the account to provide more flexibility to the users. Right now, it did not affect our progress that much because our recommendations do not rely only on Spotify information.

   (b) Get / show Spotify Info: As mentioned above, getting data is required for recommendations but displaying it on the Looking for Concerts (LFC) profiles is optional; just to prove that we are actually fetching the data.

   (c) Login with Spotify: Since we do not offer signup with Spotify (then we would have to have two separate Registered User models), a user can login with Spotify only after he or she logins once with the LFC account and connects with Spotify. Then we can match the Spotify credentials and log the user in. We currently do not have this feature but it is at the top of our priority list.

   (d) Follow/unfollow User: This is a key feature for every social platform as it builds social networks. It is good that we have it, because otherwise it would hinder our progress.

6. **Search**

    (a) Basic Search: This feature is important for the usability of our application as it enhances the navigation experience of the users. Basic search is available only for concerts at this moment but we will extend it to users as well. The fact that we did not have it in frontend did not hinder our demo but we definitely need have it as soon as possible.

    (b) Advanced Search: This feature is only available for concerts. It enables users to filter concerts according to price range, date, time, artist etc. We also received feedback to use the filtering mechanism in recommendations and we will consider it.

7. **Recommendation**

    (a) Get / show recommendations: Recommendation is the most crucial feature of Looking for Concerts. Currently we have basic recommendation just using information coming from artists. We need to consider many more sources of data to improve our system for the third milestone such as; information coming from followed users or semantic tags. Of course, our frontend and Android should also be able to display the recommendations!

8. **React Integration:** Using React on frontend enabled us to transition seamlessly into a better design and it will also help in the third milestone.

9. **Deployment:** Deploying the project is critical. We had deployed the backend for the first milestone as well, so we did not have problems there, however the frontend is still not deployed, which we expect the frontend team to do.

4 Coding work

Group Member	Team	Contribution
Haluk Alper Karaevli	Back-end	I implemented the functionalities: Recommendation; recommendation of concerts are from the artists whose concerts are attended by the user or from user's top artists according to the top artists info from spotify. Advanced and basic search; in advanced search user can search concerts with its name, artist's name, location's name, concert's tags, max and min price. In basic search, the concerts whose name, tag, location's name or artist's name contain the given string are returned. Follow/unfollow user: added necessary relation fields to registered user model and created named endpoints. Rate Concert; User rates the concert according to four categories. Subscribe/Unsubscribe concert; users can subscribe to concerts which affect their recommended concerts. Also implemented an endpoint for giving users' subscribed concerts info. Spotify search on artists; returns spotify artists (their name, spotify id and image urls) that contains given string in their names.
Kemal Berk Kocabağlı	Back-end	I implemented JWT authorization, Spotify connection and disconnection on the back-end along with a helper method to get Spotify access token given the refresh token, the skeleton of the Swagger documentation for our API and added comments to back-end where I thought was necessary. I also wrote some function based views to return user data, average concert ratings and so on. I deployed the back-end on the AWS frequently. Furthermore, I coordinated the group communication process, asking each team for their progress each week and made the oral presentation for milestone 2.
Enes Hecan	Back-end	Implemented Semantic Tag mechanism by using the Wikidata API. Designed and implemented the tag model to use acquired meaningful tags which are related with Artists, Music, and Genres with the API. Implemented send/receive functionality of images. I thought it would be robust if an actual copy of the image will remain in the database instead of keeping a URL for consistency.Hence, implemented uploading images to database. Additionally, performed Logo Design Enhancements, Bug Fixing, Code Style Improvements. Prepared slides and made preliminaries for the Milestone Demo.
Pınar Baki	Android	I implemented search functionality on the main page of the app. I created signup page and implemented signup functionality. Also, I implemented logout functionality without token authorization.
Elif Güler	Android	Learned Google Places API. Implemented artist search from Spotify via our internal API and concert location selection via Google Places API. Added a ticket link section to concert creation so users can add the link to the website which sells the concert's tickets. Implemented token logic. Revised the project plan. Created the scenario and did the demo during milestone presentation.

Group Member	Team	Contribution
Furkan Şenharputlu	Android	I implemented attend and unattend functionalities for a concert. Also, I implemented user profile page having information about user and user's concert history and future concerts. I added the functionality of staying logged in when restarting the app to login/logout part. I used shared preferences for this. Additionally, I implemented detailed rating functionality. In this way, user can rate a concert according to four different aspects. I made some theme design, colors, buttons etc.
Halil Ozan Akgül	Front-end	I made the concert detail page and the profile page. I converted my older concert detail page from Angular to React. I added some new features: "Attended", "Attending" button, the artist image, the Google Maps API to show the concert place, concert ratings and rate buttons, the tags and the "Buy" button. In the profile page, I showed the general information of user. Showed the future and past concerts and followers and the followings. Also added the follow/unfollow functionality. I also linked these pages such as when a concert is clicked user is redirected to that concert's page. Finally we used semantic-ui so I integrated my pages with that.
Alpertunga Ertin	Front-end	I have learned React for front end development and integrated React features into concert creation page, artist choosing and venue selection with Google Geocoding API are ready. I have linked profile editing page on the user profile page. I have learned getting snapshops of the react project to create static versions of our project that should be deployable on any static web hosting environment. Fixed some bugs that kept appearing on the react source code.
Oğuzhan Göller	Front-end	I created Edit Profile Page in react and integrated it with our front-end
Burak Sofu	Front-end	I learned about React and Redux and created the main file structure and Redux Store that should contain all relevent information of front-end. Re-implemented Login/Signup page, stored token and relevant info in the store to be used by other pages. Implemented main page which lists all concerts using semantic-ui as template. Implemented Top Navigation which has links to Main Page, my profile, login/signup/logout and create-a-concert page as well as search bar. Refresh access token helper function is implemented but not actively used. Validations and error handling functions, some useful helper functions that front-end developers will use are also implemented.

5 Requirements Satisfied

5.1 Functional Requirements

5.1.1 User Requirements

1.1 Registered Users

- **1.1.2** Registered users shall have a personal profile page.

  1.1.2.1 In each profile page, each user shall have a name, profile picture, age, gender and location.

  1.1.2.1.1 Users shall enter these pieces of information when signing up.

  1.1.2.1.2 Users shall be able to edit their profiles whenever they want to. **Partially, only in backend.**

  1.1.2.2 In each profile page, each user shall be able to see his concert history, following/followers, liked artists and liked genres.

1.1.2.2.1 Users shall be able to see the concerts they will attend to and the concerts they have attended in the concert history.

- **1.1.3** Registered users shall be able to follow other users.
- **1.1.4** Users shall be able to search for concert pages based on one or more of the following: artist name, venue, date, tags and location (city/district).
- **1.1.5** Users shall see and interact with concert pages.

1.1.5.1 In the concert page, users shall be able to see the artist, venue, concert date, tags, rating and comments.

1.1.5.2 Users shall be able add semantic tags about genres, artists country of origin, etc. by entering text and then choosing from the presented options retrieved from a 3rd party semantic tag repository such as Wikidata. **Partially, only in backend.**

1.1.5.3 Users shall be able to rate a concert.

1.1.5.4 Users shall be able to comment on a concert.

1.1.5.4.1 Users shall be able to upvote/downvote comments. **Partially, only in backend.**

1.1 Non-Registered Users

- **1.2.2** Non-registered users shall only be able to view concert pages and user profiles.

1.2.2.1 In the concert pages, non-registered users shall be able to see the artist, venue, concert date, tags, rating and comments.

1.2.2.3 Non-registered users shall not be able to interact with the concert pages or user profiles.

- **1.2.3** Non-registered users shall be able to search for concert pages based on one or more of the following: artist name, venue, date, tags and location (city/district).

5.2 Non-Functional Requirements

1. Usability

- **1.1** Users shall navigate to every page within the application using at most 3 links.

2. Third Party Integration

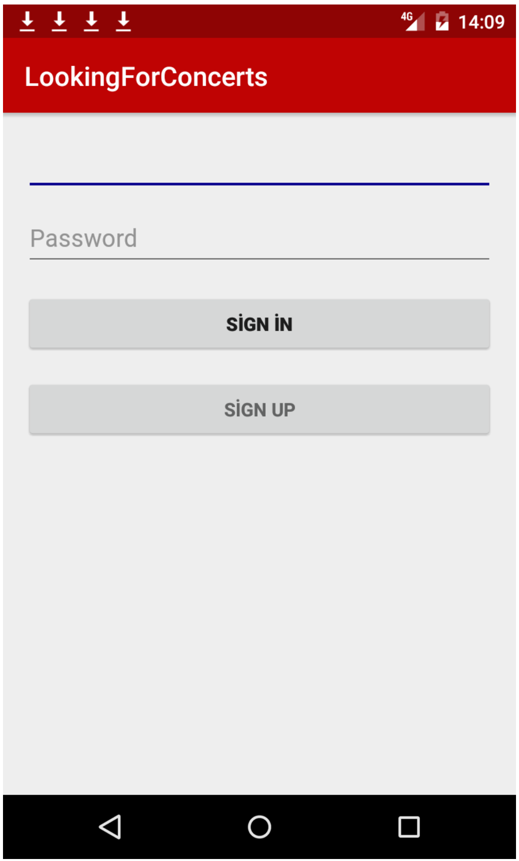
- **1.1** Google Maps API shall be used for location finding.
- **2.2** Spotify API shall be used for connecting Users to their Spotify account and giving them recommendations. **Partially.**

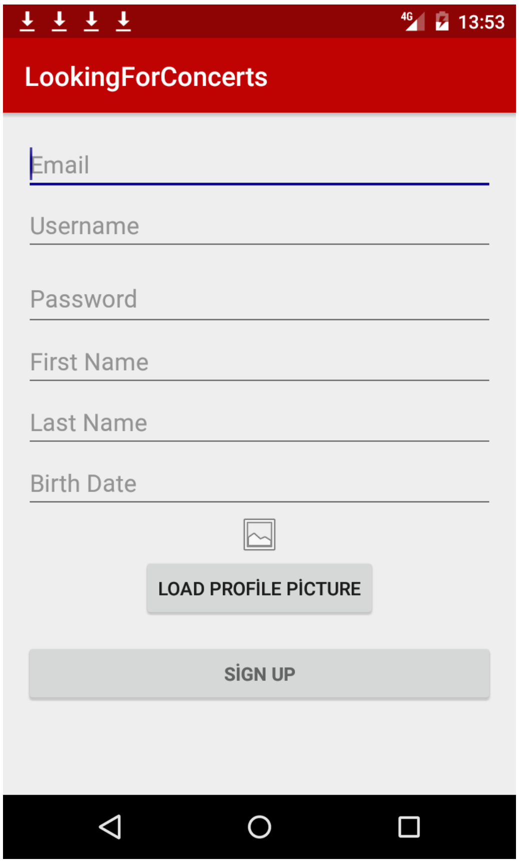
3. Security

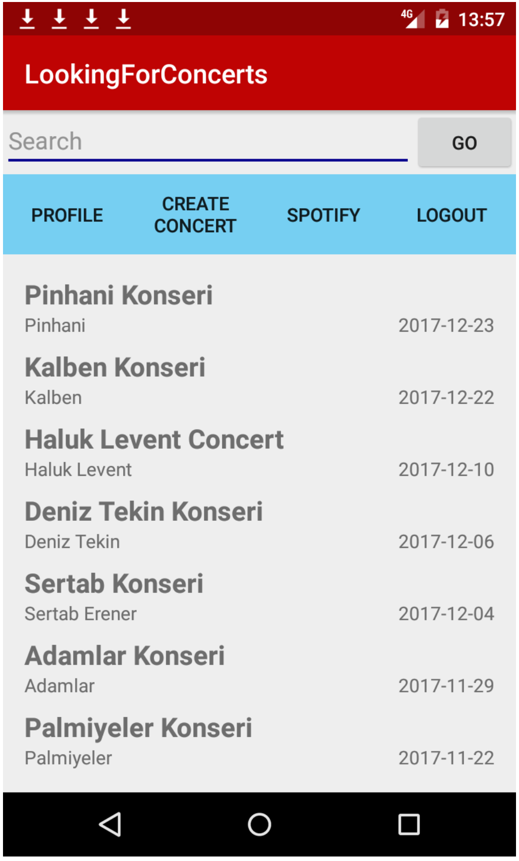
- **3.1** Database system shall be protected from unauthorized access. JWT and password hashing accomplishes this.
- **3.6** Personal info shall be kept private for each user.

6 Design

6.1 Android







Pinhani Konseri
Pinhani 2017-12-23

Kalben Konseri
Kalben 2017-12-22

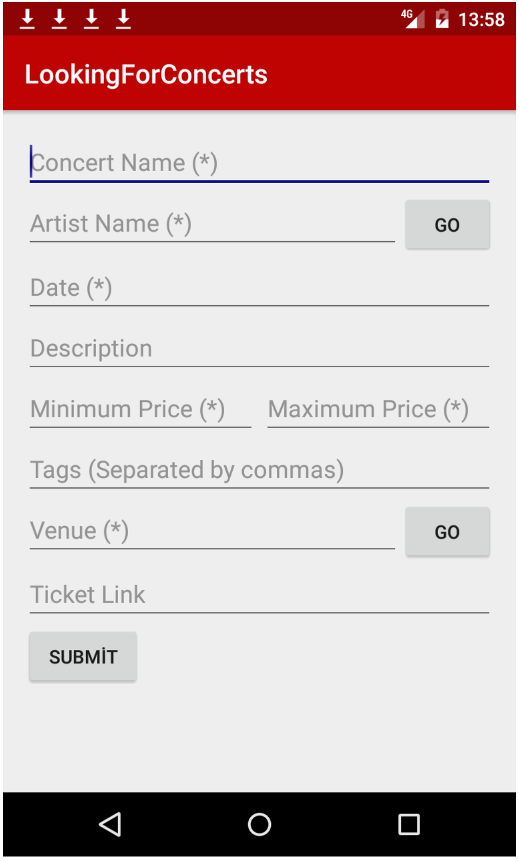
Haluk Levent Concert
Haluk Levent 2017-12-10

Deniz Tekin Konseri
Deniz Tekin 2017-12-06

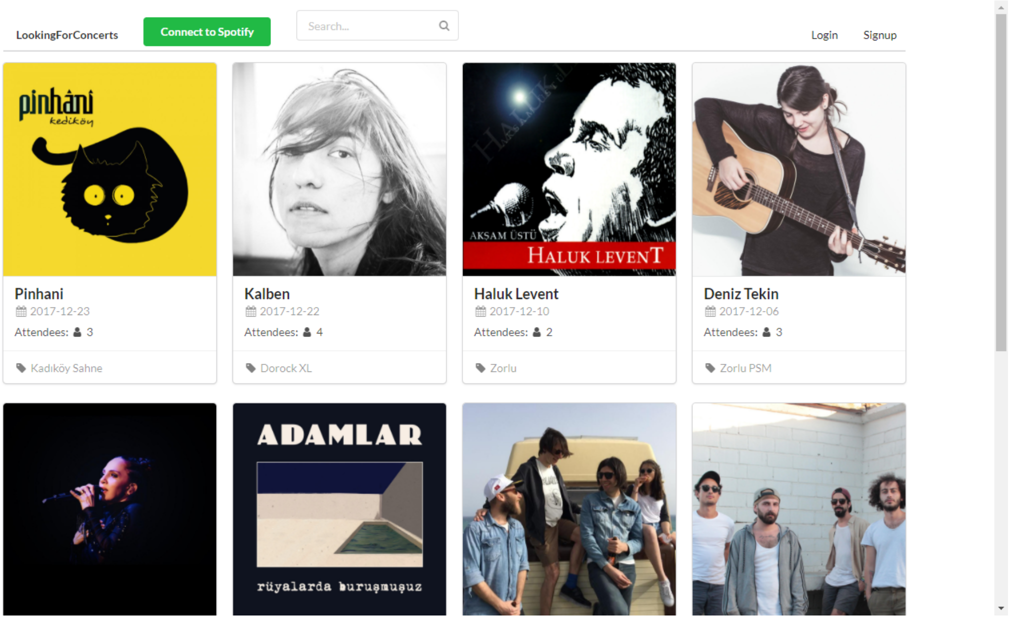
Sertab Konseri
Sertab Erener 2017-12-04

Adamlar Konseri
Adamlar 2017-11-29

Palmiyeler Konseri
Palmiyeler 2017-11-22



6.2 Web

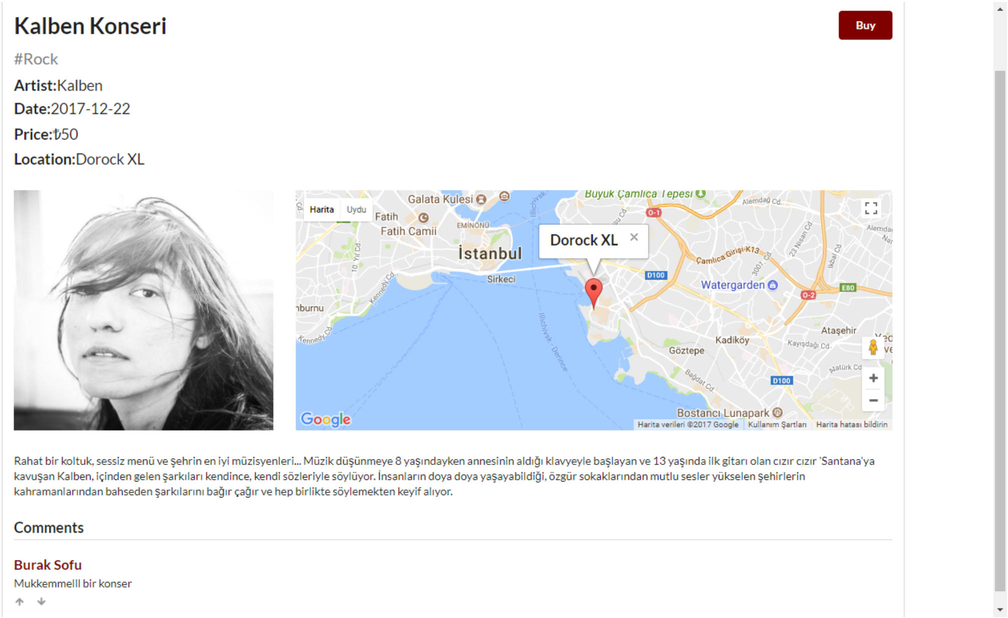


Pinhani
2017-12-23
Attendees: 3
Kadıköy Sahne

Kalben
2017-12-22
Attendees: 4
Dorock XL

Haluk Levent
2017-12-10
Attendees: 2
Zorlu

Deniz Tekin
2017-12-06
Attendees: 3
Zorlu PSM



Kalben Konseri

#Rock

Artist: Kalben

Date:2017-12-22

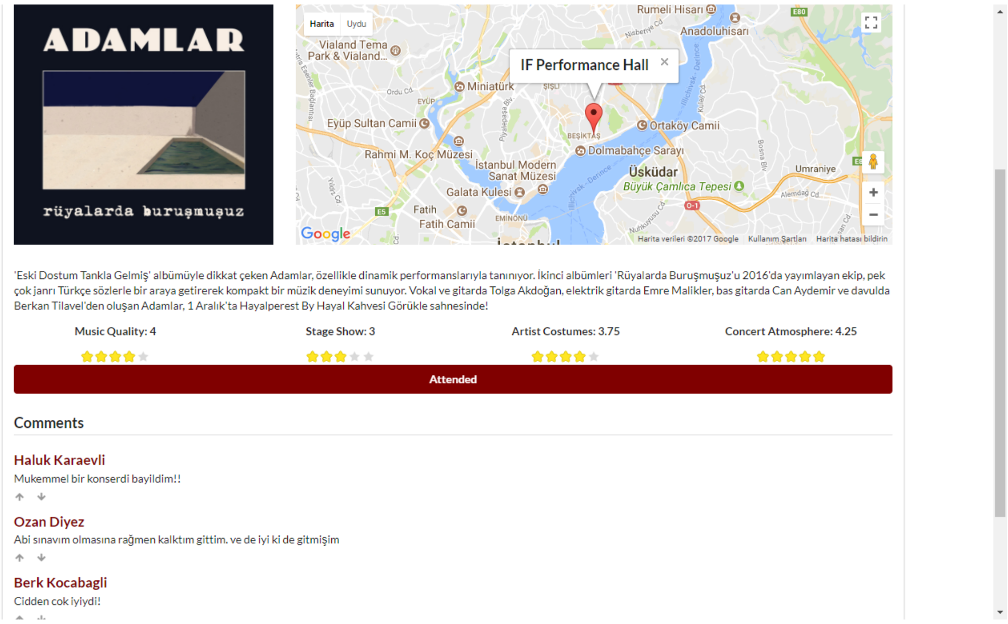
Price:₺50

Location: Dorock XL

Rahat bir koltuk, sessiz menü ve şehrin en iyi müzisyenleri... Müzik düşünmeye 8 yaşındayken annesinin aldığı klavyeyle başlayan ve 13 yaşında ilk gitarı olan cızır cızır 'Santana'ya kavuşan Kalben, içinden gelen şarkıları kendince, kendi sözleriyle söylüyor. İnsanların doya doya yaşayabildiği, özgür sokaklarından mutlu sesler yükselen şehirlerin kahramanlarından bahseden şarkılarını bağır çağır ve hep birlikte söylemekten keyif alıyor.

Comments

Burak Sofu
Mukkemmelll bir konser



'Eski Dostum Tankla Gelmiş' albümüyle dikkat çeken Adamlar, özellikle dinamik performanslarıyla tanınıyor. İkinci albümleri 'Rüyalarda Buruşmuşuz'u 2016'da yayımlayan ekip, pek çok janrı Türkçe sözlerle bir araya getirerek kompakt bir müzik deneyimi sunuyor. Vokal ve gitarda Tolga Akdoğan, elektrik gitarda Emre Malikler, bas gitarda Can Aydemir ve davulda Berkan Tilavel'den oluşan Adamlar, 1 Aralık'ta Hayalperest By Hayal Kahvesi Görükle sahnesinde!

Music Quality: 4 ★★★★☆
Stage Show: 3 ★★★☆☆
Artist Costumes: 3.75 ★★★★☆
Concert Atmosphere: 4.25 ★★★★★

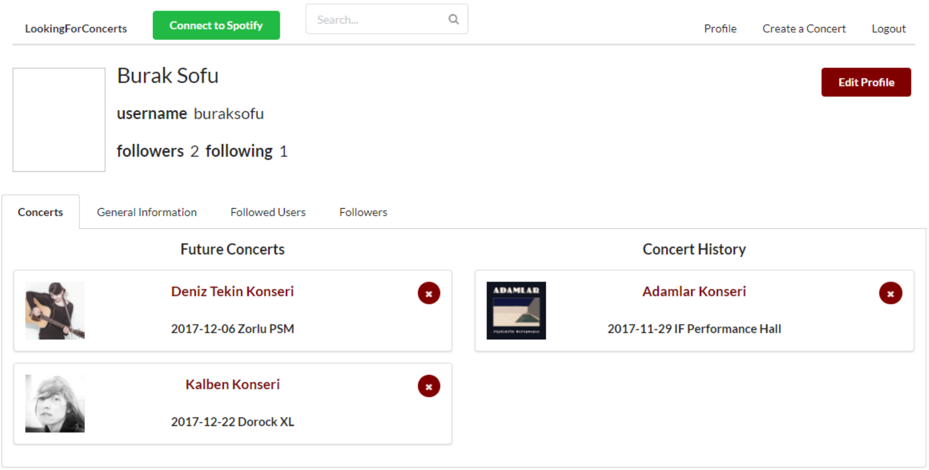
Attended

Comments

Haluk Karaevli
Mukemmel bir konserdi bayildim!!

Ozan Diyez
Abi sınavım olmasına rağmen kalktım gittim. ve de iyi ki de gitmişim

Berk Kocabagli
Cidden cok iyiydi!



Burak Sofu

username buraksofu

followers 2 **following** 1

Future Concerts

Deniz Tekin Konseri

2017-12-06 Zorlu PSM

Kalben Konseri

2017-12-22 Dorock XL

Concert History

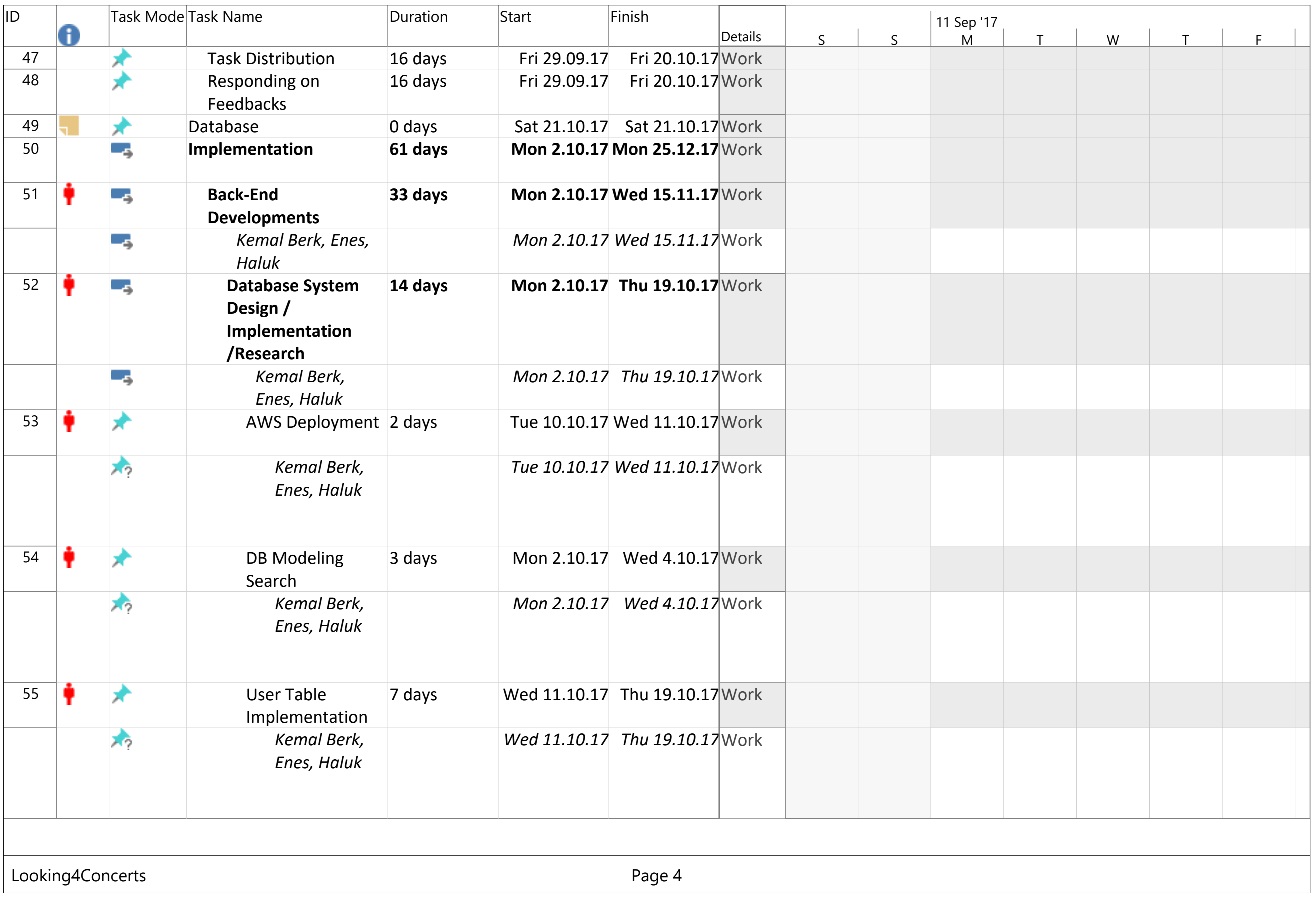
Adamlar Konseri

2017-11-29 IF Performance Hall

7 Project Plan

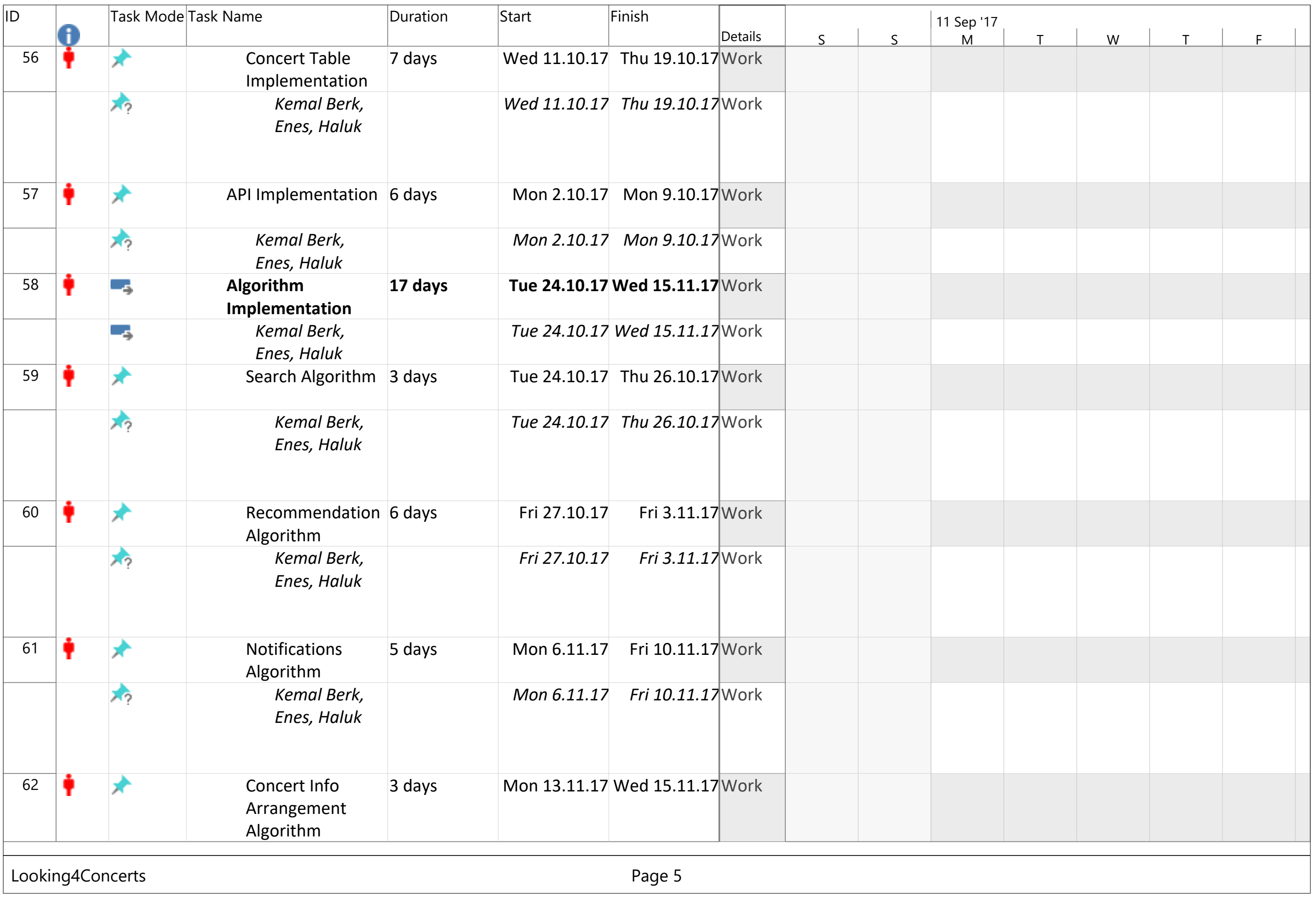
7.1 Project Plan Table

Below is the part of our project plan for this semester.



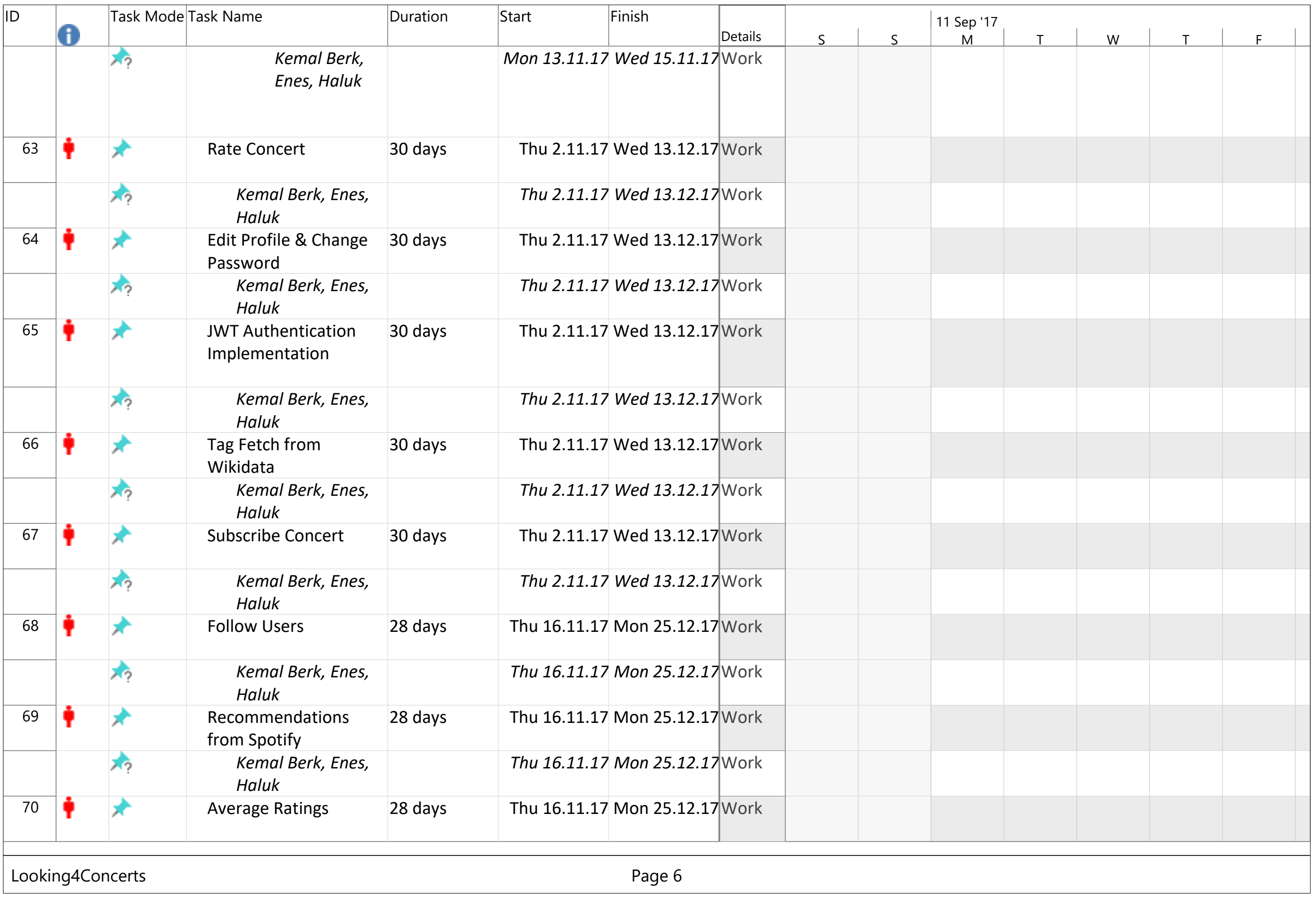
ID		Task Mode	Task Name	Duration	Start	Finish	Details	S	S	11 Sep '17 M	T	W	T	F
47			Task Distribution	16 days	Fri 29.09.17	Fri 20.10.17	Work							
48			Responding on Feedbacks	16 days	Fri 29.09.17	Fri 20.10.17	Work							
49			Database	0 days	Sat 21.10.17	Sat 21.10.17	Work							
50			**Implementation**	**61 days**	**Mon 2.10.17**	**Mon 25.12.17**	Work							
51			**Back-End Developments**	**33 days**	**Mon 2.10.17**	**Wed 15.11.17**	Work							
			Kemal Berk, Enes, Haluk		*Mon 2.10.17*	*Wed 15.11.17*	Work							
52			**Database System Design / Implementation /Research**	**14 days**	**Mon 2.10.17**	**Thu 19.10.17**	Work							
			Kemal Berk, Enes, Haluk		*Mon 2.10.17*	*Thu 19.10.17*	Work							
53			AWS Deployment	2 days	Tue 10.10.17	Wed 11.10.17	Work							
			Kemal Berk, Enes, Haluk		*Tue 10.10.17*	*Wed 11.10.17*	Work							
54			DB Modeling Search	3 days	Mon 2.10.17	Wed 4.10.17	Work							
			Kemal Berk, Enes, Haluk		*Mon 2.10.17*	*Wed 4.10.17*	Work							
55			User Table Implementation	7 days	Wed 11.10.17	Thu 19.10.17	Work							
			Kemal Berk, Enes, Haluk		*Wed 11.10.17*	*Thu 19.10.17*	Work							

Looking4Concerts Page 4



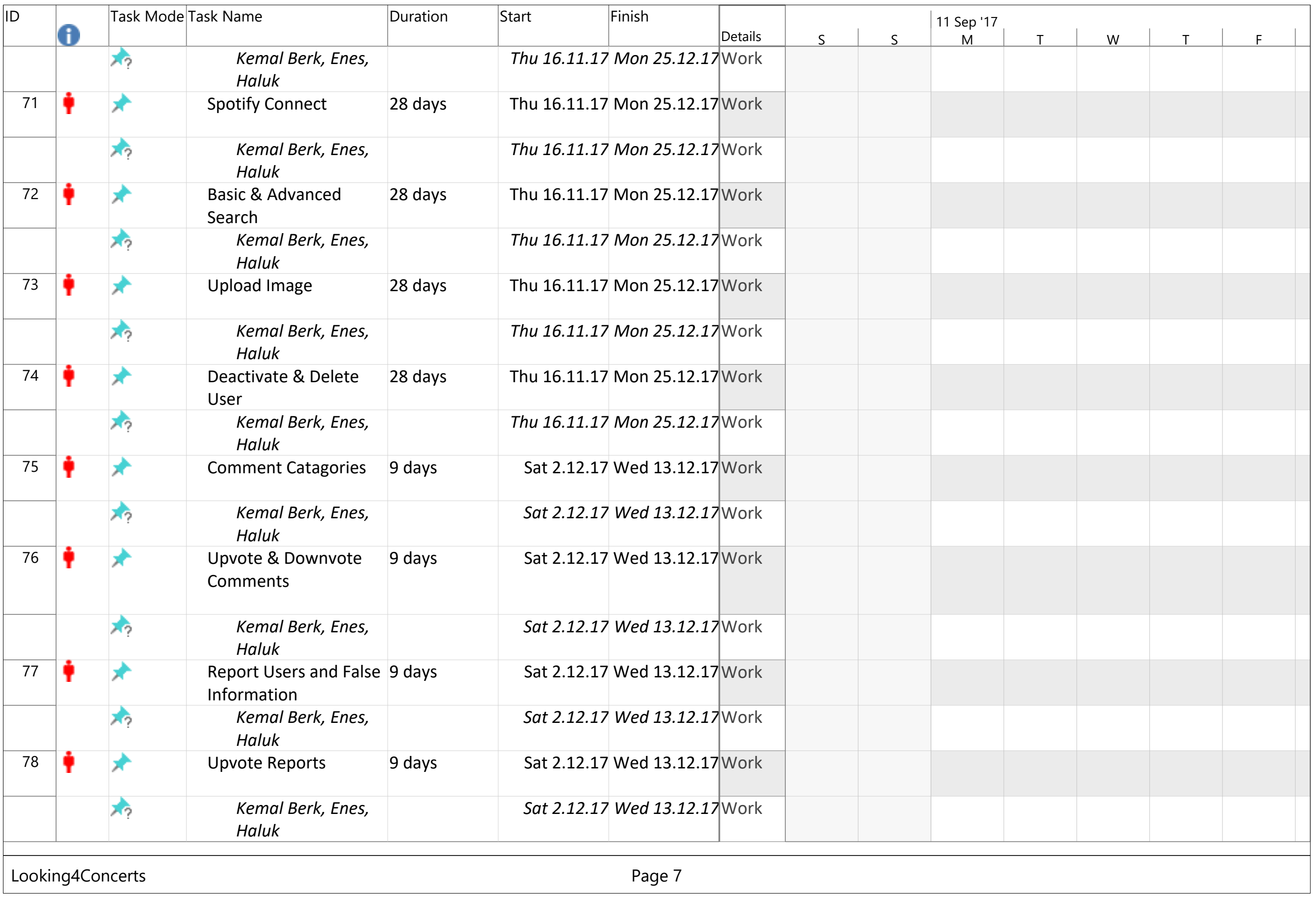
ID		Task Mode	Task Name	Duration	Start	Finish	Details	S	S	11 Sep '17 M	T	W	T	F
56			Concert Table Implementation	7 days	Wed 11.10.17	Thu 19.10.17	Work							
			Kemal Berk, Enes, Haluk		*Wed 11.10.17*	*Thu 19.10.17*	Work							
57			API Implementation	6 days	Mon 2.10.17	Mon 9.10.17	Work							
			Kemal Berk, Enes, Haluk		*Mon 2.10.17*	*Mon 9.10.17*	Work							
58			**Algorithm Implementation**	**17 days**	**Tue 24.10.17**	**Wed 15.11.17**	Work							
			Kemal Berk, Enes, Haluk		*Tue 24.10.17*	*Wed 15.11.17*	Work							
59			Search Algorithm	3 days	Tue 24.10.17	Thu 26.10.17	Work							
			Kemal Berk, Enes, Haluk		*Tue 24.10.17*	*Thu 26.10.17*	Work							
60			Recommendation Algorithm	6 days	Fri 27.10.17	Fri 3.11.17	Work							
			Kemal Berk, Enes, Haluk		*Fri 27.10.17*	*Fri 3.11.17*	Work							
61			Notifications Algorithm	5 days	Mon 6.11.17	Fri 10.11.17	Work							
			Kemal Berk, Enes, Haluk		*Mon 6.11.17*	*Fri 10.11.17*	Work							
62			Concert Info Arrangement Algorithm	3 days	Mon 13.11.17	Wed 15.11.17	Work							

Looking4Concerts Page 5



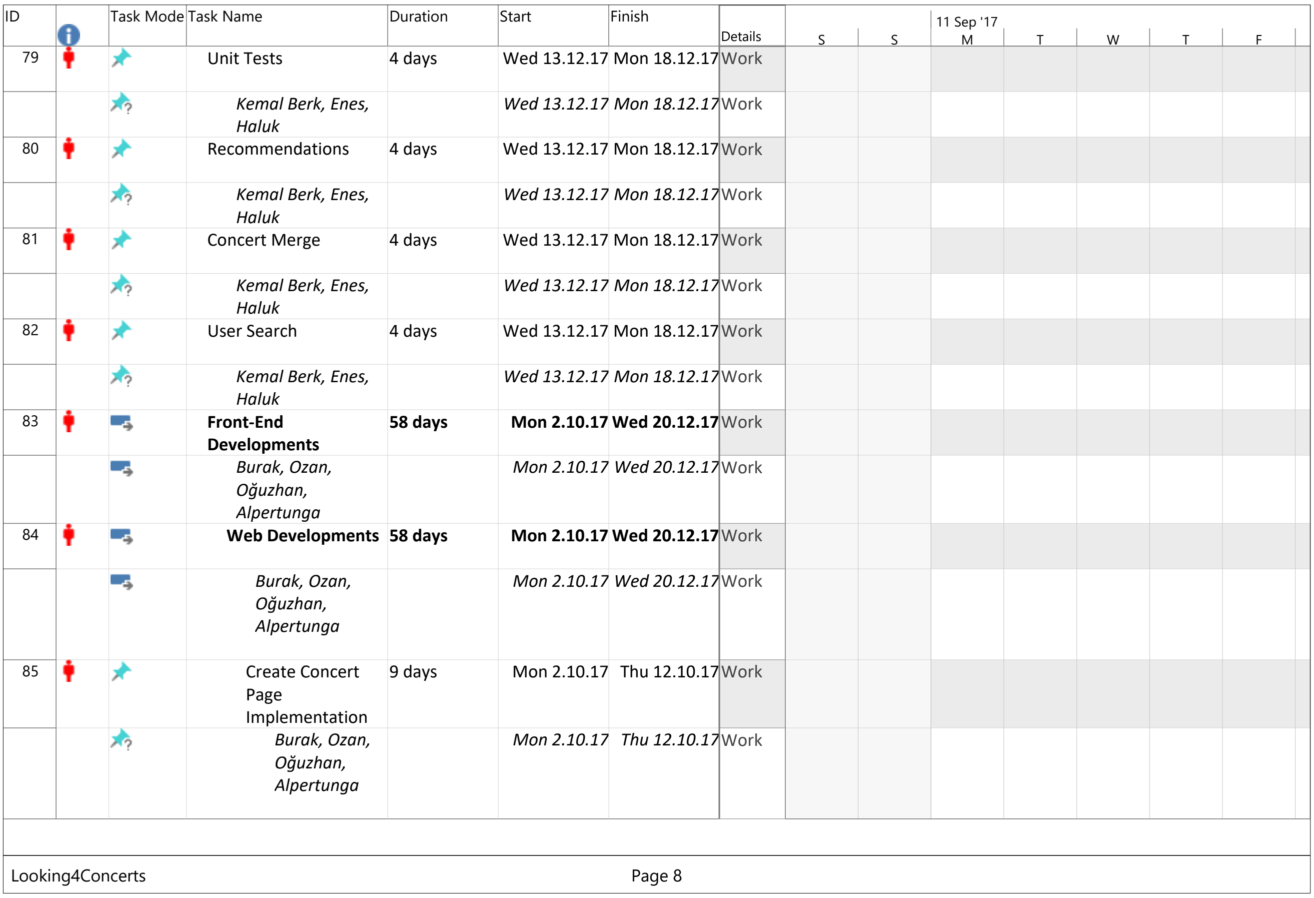
ID	ℹ	Task Mode	Task Name	Duration	Start	Finish	Details	S	S	11 Sep '17 M	T	W	T	F
		📌?	*Kemal Berk, Enes, Haluk*		*Mon 13.11.17*	*Wed 15.11.17*	Work							
63	👤	📌	Rate Concert	30 days	Thu 2.11.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 2.11.17*	*Wed 13.12.17*	Work							
64	👤	📌	Edit Profile & Change Password	30 days	Thu 2.11.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 2.11.17*	*Wed 13.12.17*	Work							
65	👤	📌	JWT Authentication Implementation	30 days	Thu 2.11.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 2.11.17*	*Wed 13.12.17*	Work							
66	👤	📌	Tag Fetch from Wikidata	30 days	Thu 2.11.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 2.11.17*	*Wed 13.12.17*	Work							
67	👤	📌	Subscribe Concert	30 days	Thu 2.11.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 2.11.17*	*Wed 13.12.17*	Work							
68	👤	📌	Follow Users	28 days	Thu 16.11.17	Mon 25.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 16.11.17*	*Mon 25.12.17*	Work							
69	👤	📌	Recommendations from Spotify	28 days	Thu 16.11.17	Mon 25.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 16.11.17*	*Mon 25.12.17*	Work							
70	👤	📌	Average Ratings	28 days	Thu 16.11.17	Mon 25.12.17	Work							

Looking4Concerts Page 6



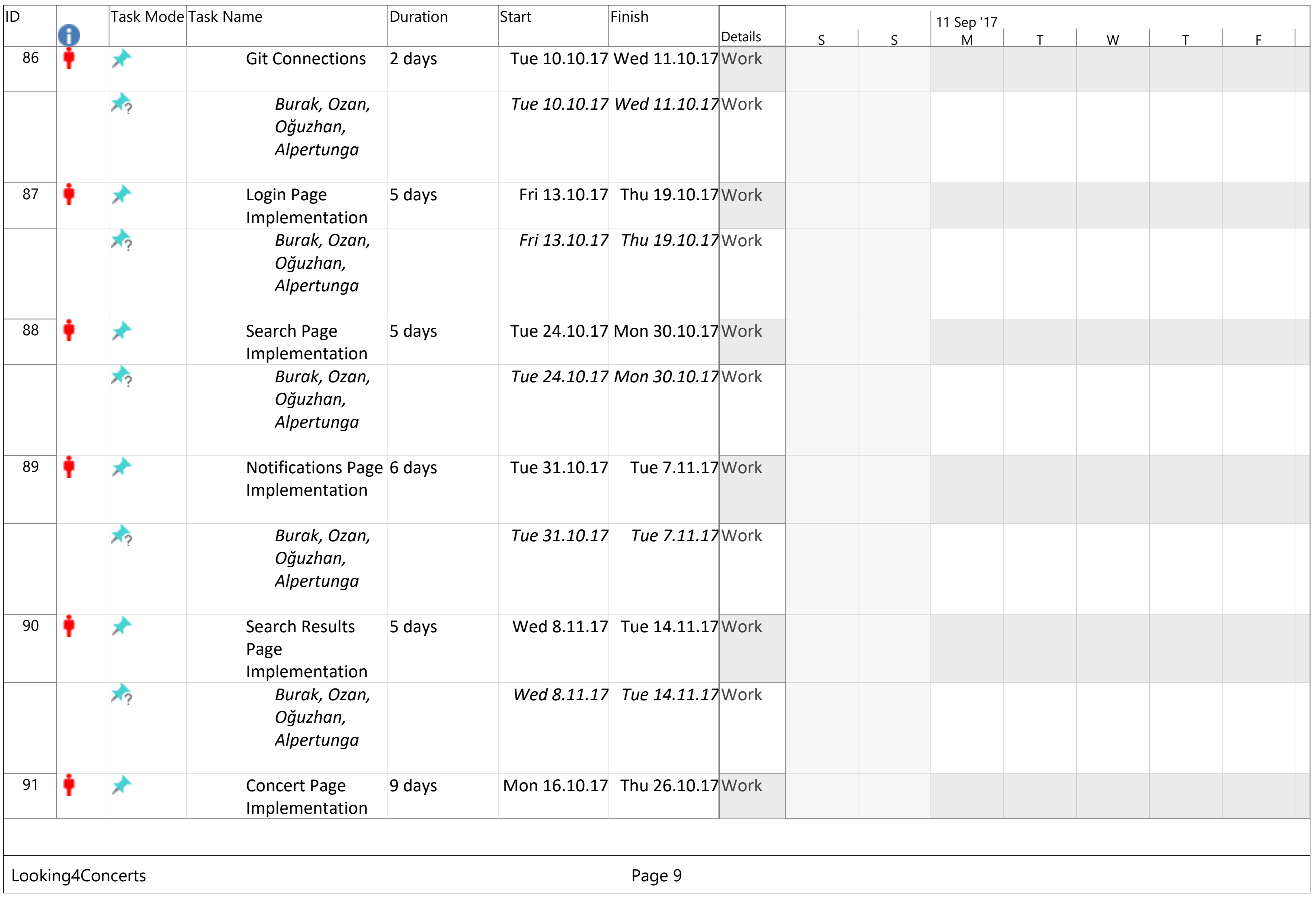
ID	ℹ	Task Mode	Task Name	Duration	Start	Finish	Details	S	S	11 Sep '17 M	T	W	T	F
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 16.11.17*	*Mon 25.12.17*	Work							
71	👤	📌	Spotify Connect	28 days	Thu 16.11.17	Mon 25.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 16.11.17*	*Mon 25.12.17*	Work							
72	👤	📌	Basic & Advanced Search	28 days	Thu 16.11.17	Mon 25.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 16.11.17*	*Mon 25.12.17*	Work							
73	👤	📌	Upload Image	28 days	Thu 16.11.17	Mon 25.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 16.11.17*	*Mon 25.12.17*	Work							
74	👤	📌	Deactivate & Delete User	28 days	Thu 16.11.17	Mon 25.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Thu 16.11.17*	*Mon 25.12.17*	Work							
75	👤	📌	Comment Catagories	9 days	Sat 2.12.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Sat 2.12.17*	*Wed 13.12.17*	Work							
76	👤	📌	Upvote & Downvote Comments	9 days	Sat 2.12.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Sat 2.12.17*	*Wed 13.12.17*	Work							
77	👤	📌	Report Users and False Information	9 days	Sat 2.12.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Sat 2.12.17*	*Wed 13.12.17*	Work							
78	👤	📌	Upvote Reports	9 days	Sat 2.12.17	Wed 13.12.17	Work							
		📌?	*Kemal Berk, Enes, Haluk*		*Sat 2.12.17*	*Wed 13.12.17*	Work							

Looking4Concerts Page 7



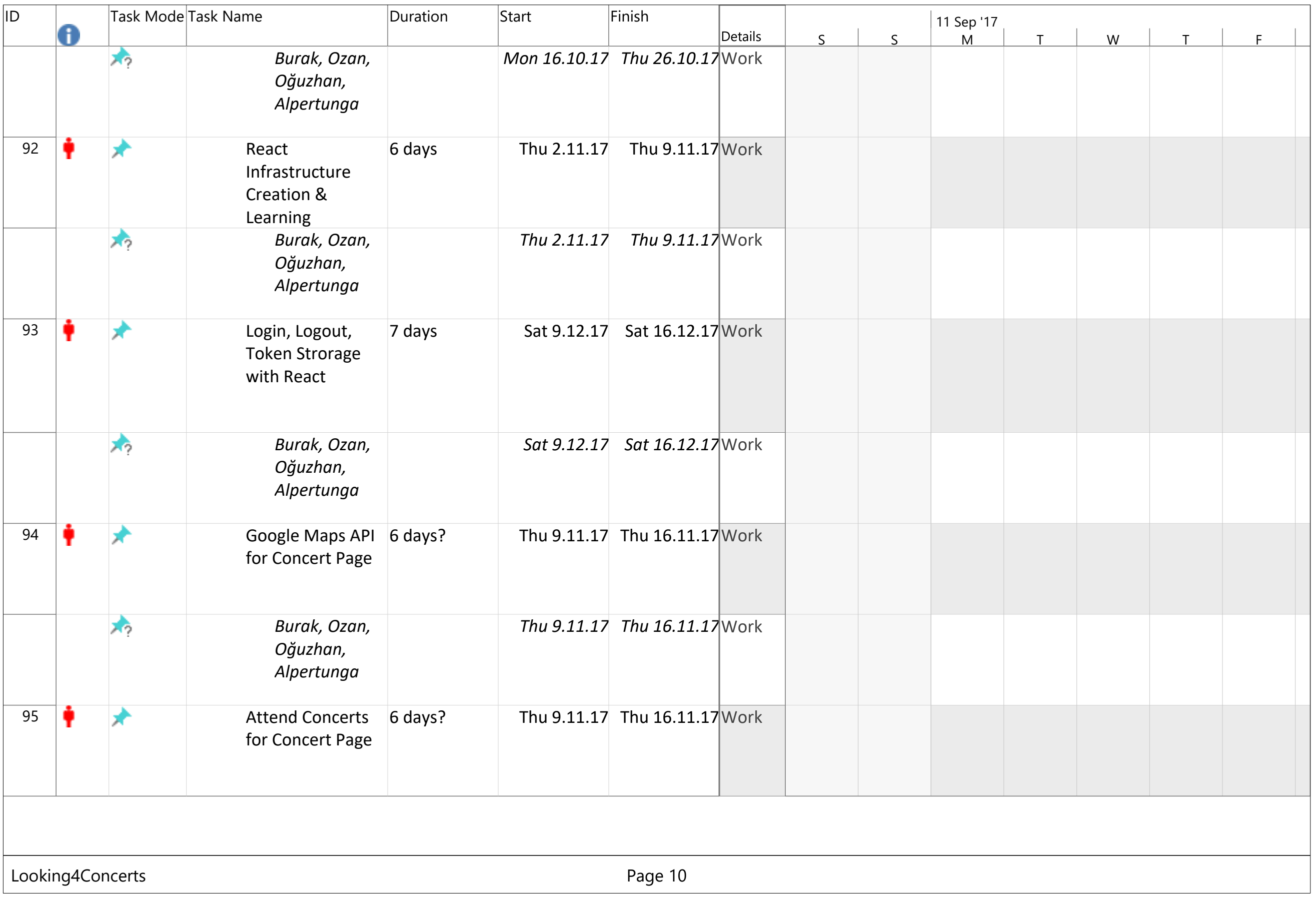
ID		Task Mode	Task Name	Duration	Start	Finish	Details	S	S	11 Sep '17 M	T	W	T	F
79			Unit Tests	4 days	Wed 13.12.17	Mon 18.12.17	Work							
			Kemal Berk, Enes, Haluk		*Wed 13.12.17*	*Mon 18.12.17*	Work							
80			Recommendations	4 days	Wed 13.12.17	Mon 18.12.17	Work							
			Kemal Berk, Enes, Haluk		*Wed 13.12.17*	*Mon 18.12.17*	Work							
81			Concert Merge	4 days	Wed 13.12.17	Mon 18.12.17	Work							
			Kemal Berk, Enes, Haluk		*Wed 13.12.17*	*Mon 18.12.17*	Work							
82			User Search	4 days	Wed 13.12.17	Mon 18.12.17	Work							
			Kemal Berk, Enes, Haluk		*Wed 13.12.17*	*Mon 18.12.17*	Work							
83			**Front-End Developments**	**58 days**	**Mon 2.10.17**	**Wed 20.12.17**	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Mon 2.10.17*	*Wed 20.12.17*	Work							
84			**Web Developments**	**58 days**	**Mon 2.10.17**	**Wed 20.12.17**	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Mon 2.10.17*	*Wed 20.12.17*	Work							
85			Create Concert Page Implementation	9 days	Mon 2.10.17	Thu 12.10.17	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Mon 2.10.17*	*Thu 12.10.17*	Work							

Looking4Concerts Page 8



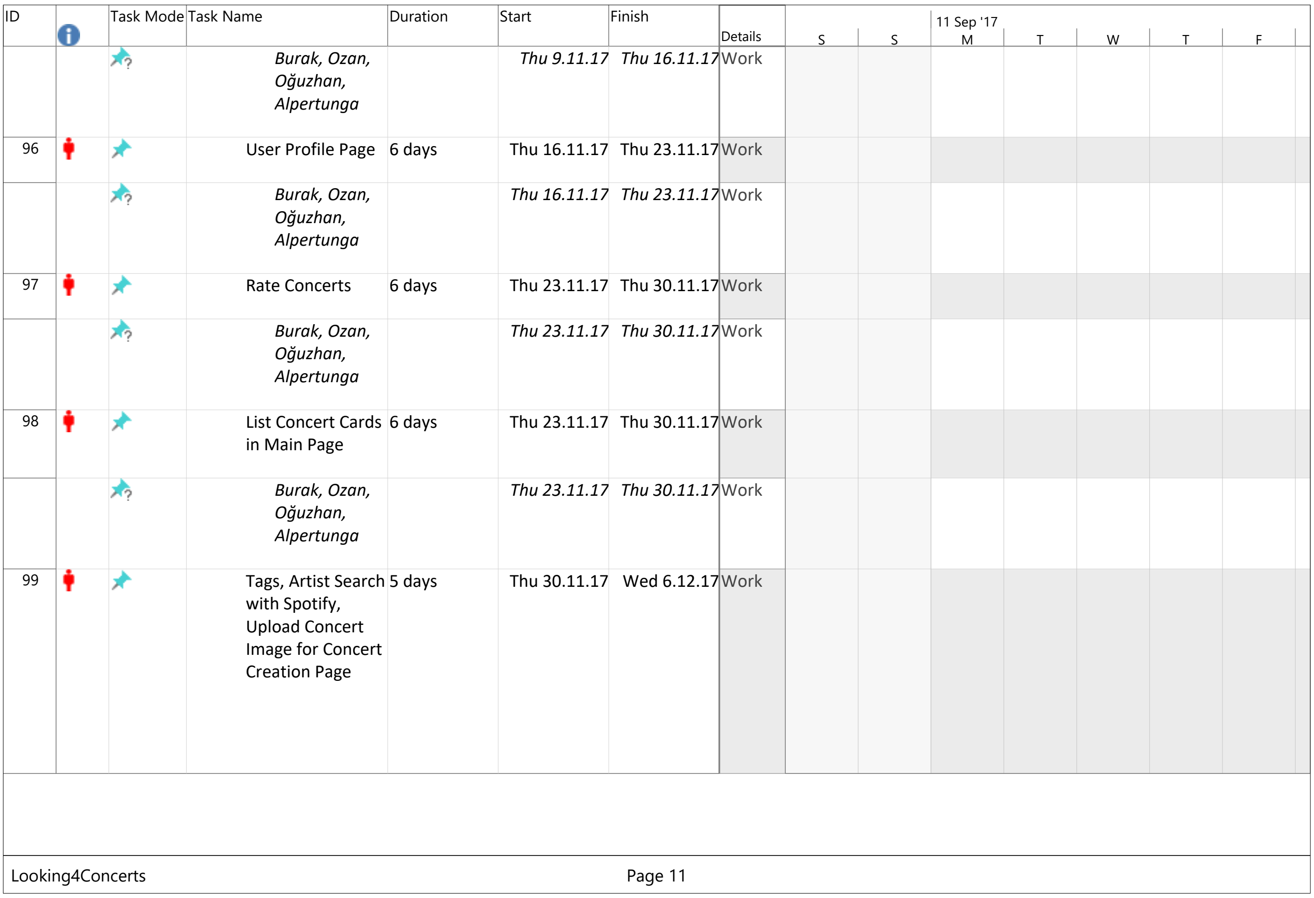
ID		Task Mode	Task Name	Duration	Start	Finish	Details	S	S	11 Sep '17 M	T	W	T	F
86			Git Connections	2 days	Tue 10.10.17	Wed 11.10.17	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Tue 10.10.17*	*Wed 11.10.17*	Work							
87			Login Page Implementation	5 days	Fri 13.10.17	Thu 19.10.17	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Fri 13.10.17*	*Thu 19.10.17*	Work							
88			Search Page Implementation	5 days	Tue 24.10.17	Mon 30.10.17	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Tue 24.10.17*	*Mon 30.10.17*	Work							
89			Notifications Page Implementation	6 days	Tue 31.10.17	Tue 7.11.17	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Tue 31.10.17*	*Tue 7.11.17*	Work							
90			Search Results Page Implementation	5 days	Wed 8.11.17	Tue 14.11.17	Work							
			Burak, Ozan, Oğuzhan, Alpertunga		*Wed 8.11.17*	*Tue 14.11.17*	Work							
91			Concert Page Implementation	9 days	Mon 16.10.17	Thu 26.10.17	Work							

Looking4Concerts Page 9



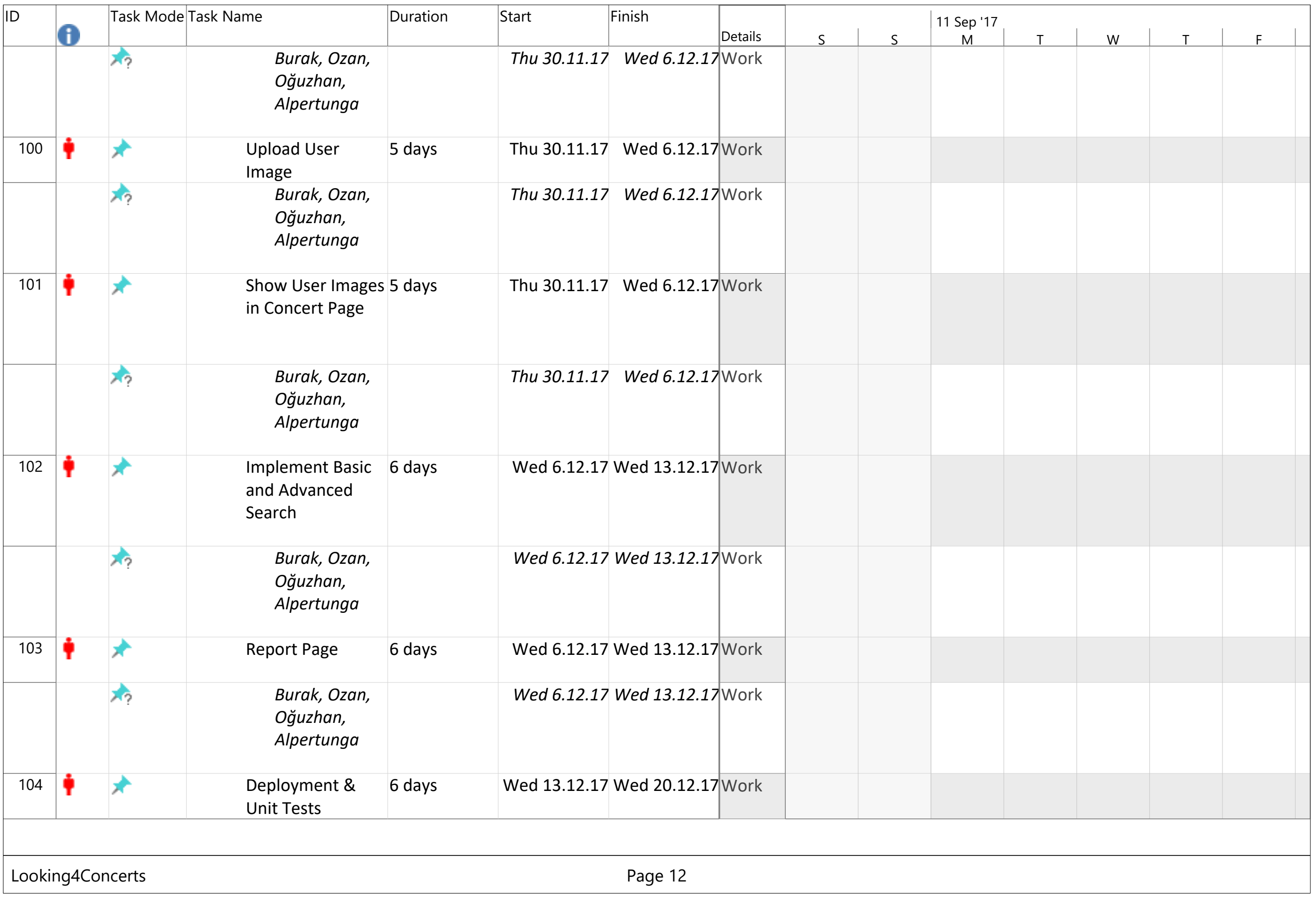
ID		Task Mode	Task Name	Duration	Start	Finish	Details			11 Sep '17				
								S	S	M	T	W	T	F
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Mon 16.10.17*	*Thu 26.10.17*	Work							
92	👤	📌	React Infrastructure Creation & Learning	6 days	Thu 2.11.17	Thu 9.11.17	Work							
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Thu 2.11.17*	*Thu 9.11.17*	Work							
93	👤	📌	Login, Logout, Token Strorage with React	7 days	Sat 9.12.17	Sat 16.12.17	Work							
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Sat 9.12.17*	*Sat 16.12.17*	Work							
94	👤	📌	Google Maps API for Concert Page	6 days?	Thu 9.11.17	Thu 16.11.17	Work							
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Thu 9.11.17*	*Thu 16.11.17*	Work							
95	👤	📌	Attend Concerts for Concert Page	6 days?	Thu 9.11.17	Thu 16.11.17	Work							

Looking4Concerts Page 10



ID		Task Mode	Task Name	Duration	Start	Finish	Details			11 Sep '17				
								S	S	M	T	W	T	F
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Thu 9.11.17*	*Thu 16.11.17*	Work							
96	👤	📌	User Profile Page	6 days	Thu 16.11.17	Thu 23.11.17	Work							
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Thu 16.11.17*	*Thu 23.11.17*	Work							
97	👤	📌	Rate Concerts	6 days	Thu 23.11.17	Thu 30.11.17	Work							
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Thu 23.11.17*	*Thu 30.11.17*	Work							
98	👤	📌	List Concert Cards in Main Page	6 days	Thu 23.11.17	Thu 30.11.17	Work							
		📌?	*Burak, Ozan, Oğuzhan, Alpertunga*		*Thu 23.11.17*	*Thu 30.11.17*	Work							
99	👤	📌	Tags, Artist Search with Spotify, Upload Concert Image for Concert Creation Page	5 days	Thu 30.11.17	Wed 6.12.17	Work							

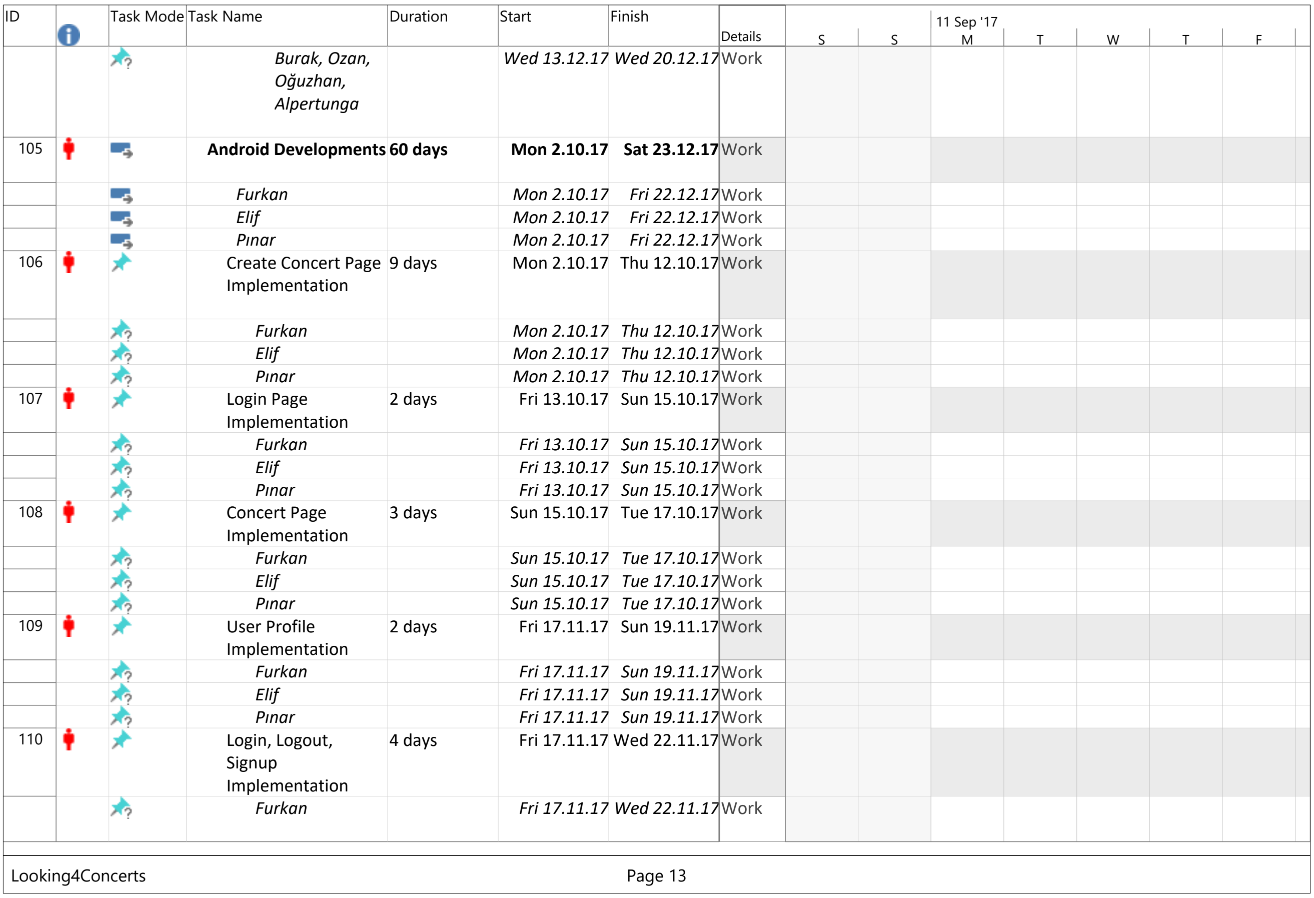
Looking4Concerts Page 11



ID	Task Mode	Task Name	Duration	Start	Finish	Details
		Burak, Ozan, Oğuzhan, Alpertunga		*Thu 30.11.17*	*Wed 6.12.17*	Work
100		Upload User Image	5 days	Thu 30.11.17	Wed 6.12.17	Work
		Burak, Ozan, Oğuzhan, Alpertunga		*Thu 30.11.17*	*Wed 6.12.17*	Work
101		Show User Images in Concert Page	5 days	Thu 30.11.17	Wed 6.12.17	Work
		Burak, Ozan, Oğuzhan, Alpertunga		*Thu 30.11.17*	*Wed 6.12.17*	Work
102		Implement Basic and Advanced Search	6 days	Wed 6.12.17	Wed 13.12.17	Work
		Burak, Ozan, Oğuzhan, Alpertunga		*Wed 6.12.17*	*Wed 13.12.17*	Work
103		Report Page	6 days	Wed 6.12.17	Wed 13.12.17	Work
		Burak, Ozan, Oğuzhan, Alpertunga		*Wed 6.12.17*	*Wed 13.12.17*	Work
104		Deployment & Unit Tests	6 days	Wed 13.12.17	Wed 20.12.17	Work

11 Sep '17 — S S M T W T F

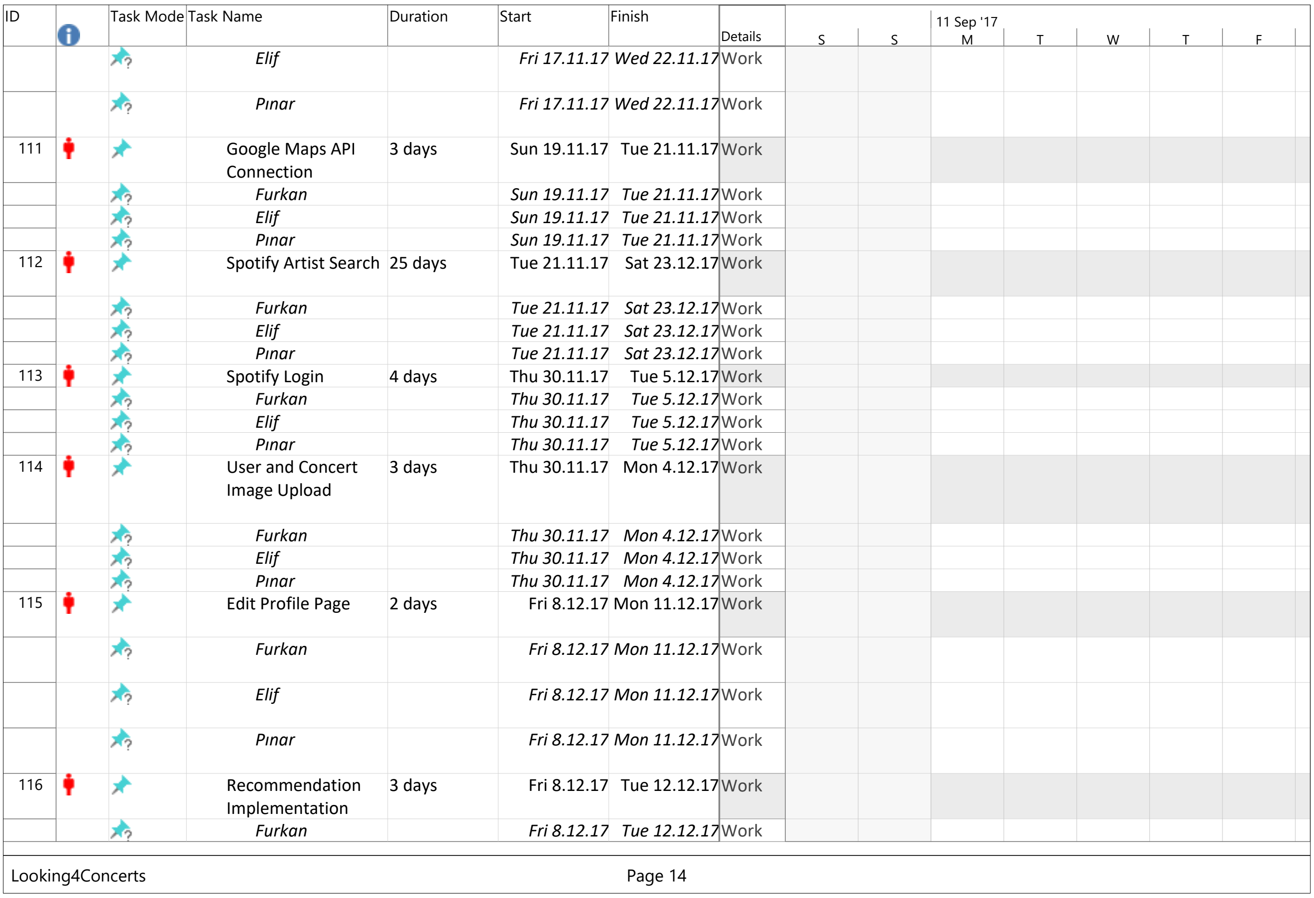
Looking4Concerts Page 12



ID	Task Mode	Task Name	Duration	Start	Finish	Details
		Burak, Ozan, Oğuzhan, Alpertunga		*Wed 13.12.17*	*Wed 20.12.17*	Work
105		**Android Developments**	**60 days**	**Mon 2.10.17**	**Sat 23.12.17**	Work
		Furkan		*Mon 2.10.17*	*Fri 22.12.17*	Work
		Elif		*Mon 2.10.17*	*Fri 22.12.17*	Work
		Pınar		*Mon 2.10.17*	*Fri 22.12.17*	Work
106		Create Concert Page Implementation	9 days	Mon 2.10.17	Thu 12.10.17	Work
		Furkan		*Mon 2.10.17*	*Thu 12.10.17*	Work
		Elif		*Mon 2.10.17*	*Thu 12.10.17*	Work
		Pınar		*Mon 2.10.17*	*Thu 12.10.17*	Work
107		Login Page Implementation	2 days	Fri 13.10.17	Sun 15.10.17	Work
		Furkan		*Fri 13.10.17*	*Sun 15.10.17*	Work
		Elif		*Fri 13.10.17*	*Sun 15.10.17*	Work
		Pınar		*Fri 13.10.17*	*Sun 15.10.17*	Work
108		Concert Page Implementation	3 days	Sun 15.10.17	Tue 17.10.17	Work
		Furkan		*Sun 15.10.17*	*Tue 17.10.17*	Work
		Elif		*Sun 15.10.17*	*Tue 17.10.17*	Work
		Pınar		*Sun 15.10.17*	*Tue 17.10.17*	Work
109		User Profile Implementation	2 days	Fri 17.11.17	Sun 19.11.17	Work
		Furkan		*Fri 17.11.17*	*Sun 19.11.17*	Work
		Elif		*Fri 17.11.17*	*Sun 19.11.17*	Work
		Pınar		*Fri 17.11.17*	*Sun 19.11.17*	Work
110		Login, Logout, Signup Implementation	4 days	Fri 17.11.17	Wed 22.11.17	Work
		Furkan		*Fri 17.11.17*	*Wed 22.11.17*	Work

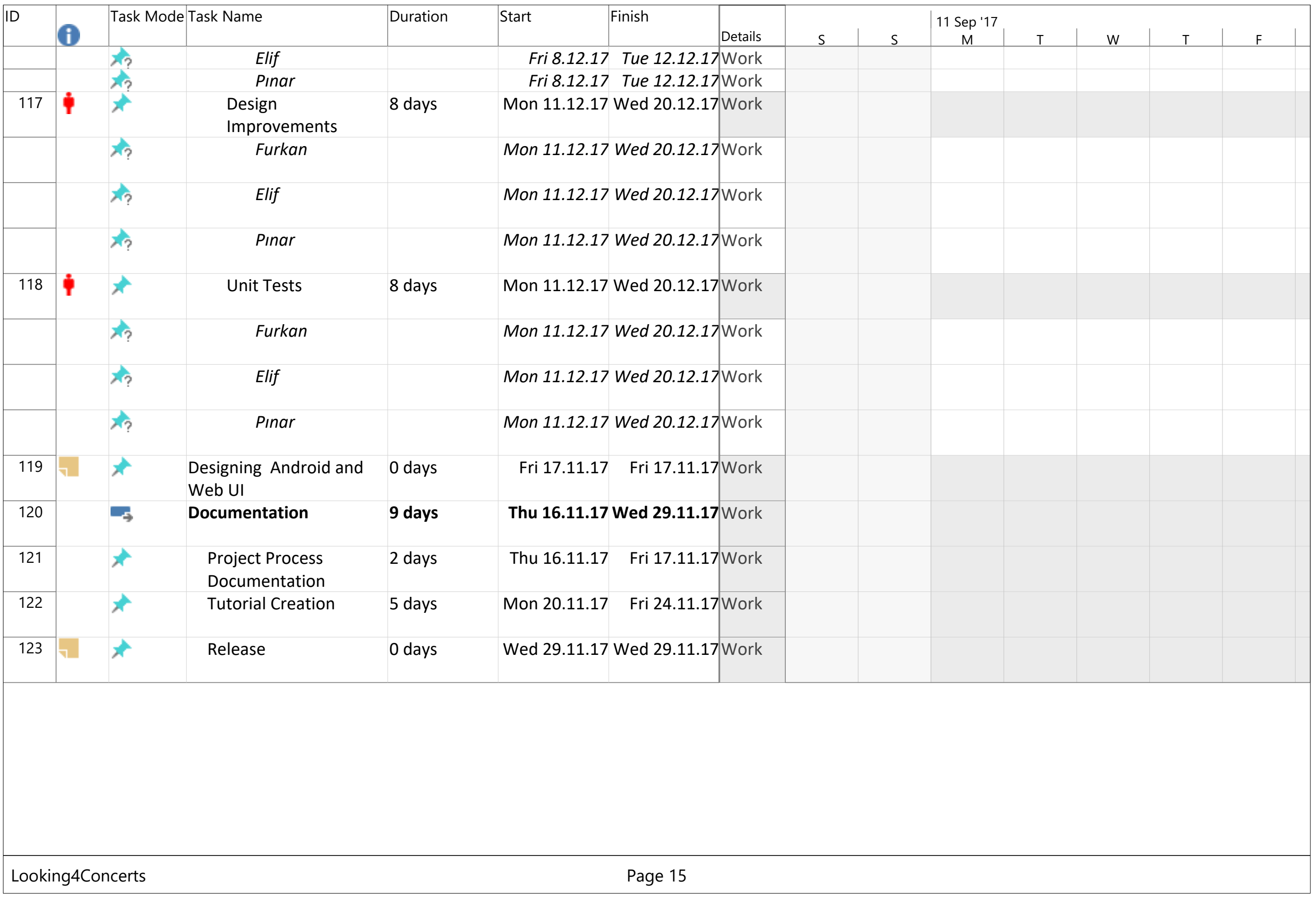
11 Sep '17 — S S M T W T F

Looking4Concerts Page 13



ID		Task Mode	Task Name	Duration	Start	Finish	Details			11 Sep '17				
								S	S	M	T	W	T	F
			Elif		*Fri 17.11.17*	*Wed 22.11.17*	Work							
			Pınar		*Fri 17.11.17*	*Wed 22.11.17*	Work							
111			Google Maps API Connection	3 days	Sun 19.11.17	Tue 21.11.17	Work							
			Furkan		*Sun 19.11.17*	*Tue 21.11.17*	Work							
			Elif		*Sun 19.11.17*	*Tue 21.11.17*	Work							
			Pınar		*Sun 19.11.17*	*Tue 21.11.17*	Work							
112			Spotify Artist Search	25 days	Tue 21.11.17	Sat 23.12.17	Work							
			Furkan		*Tue 21.11.17*	*Sat 23.12.17*	Work							
			Elif		*Tue 21.11.17*	*Sat 23.12.17*	Work							
			Pınar		*Tue 21.11.17*	*Sat 23.12.17*	Work							
113			Spotify Login	4 days	Thu 30.11.17	Tue 5.12.17	Work							
			Furkan		*Thu 30.11.17*	*Tue 5.12.17*	Work							
			Elif		*Thu 30.11.17*	*Tue 5.12.17*	Work							
			Pınar		*Thu 30.11.17*	*Tue 5.12.17*	Work							
114			User and Concert Image Upload	3 days	Thu 30.11.17	Mon 4.12.17	Work							
			Furkan		*Thu 30.11.17*	*Mon 4.12.17*	Work							
			Elif		*Thu 30.11.17*	*Mon 4.12.17*	Work							
			Pınar		*Thu 30.11.17*	*Mon 4.12.17*	Work							
115			Edit Profile Page	2 days	Fri 8.12.17	Mon 11.12.17	Work							
			Furkan		*Fri 8.12.17*	*Mon 11.12.17*	Work							
			Elif		*Fri 8.12.17*	*Mon 11.12.17*	Work							
			Pınar		*Fri 8.12.17*	*Mon 11.12.17*	Work							
116			Recommendation Implementation	3 days	Fri 8.12.17	Tue 12.12.17	Work							
			Furkan		*Fri 8.12.17*	*Tue 12.12.17*	Work							

Looking4Concerts Page 14



ID		Task Mode	Task Name	Duration	Start	Finish	Details			11 Sep '17				
								S	S	M	T	W	T	F
			Elif		*Fri 8.12.17*	*Tue 12.12.17*	Work							
			Pınar		*Fri 8.12.17*	*Tue 12.12.17*	Work							
117			Design Improvements	8 days	Mon 11.12.17	Wed 20.12.17	Work							
			Furkan		*Mon 11.12.17*	*Wed 20.12.17*	Work							
			Elif		*Mon 11.12.17*	*Wed 20.12.17*	Work							
			Pınar		*Mon 11.12.17*	*Wed 20.12.17*	Work							
118			Unit Tests	8 days	Mon 11.12.17	Wed 20.12.17	Work							
			Furkan		*Mon 11.12.17*	*Wed 20.12.17*	Work							
			Elif		*Mon 11.12.17*	*Wed 20.12.17*	Work							
			Pınar		*Mon 11.12.17*	*Wed 20.12.17*	Work							
119			Designing Android and Web UI	0 days	Fri 17.11.17	Fri 17.11.17	Work							
120			**Documentation**	**9 days**	**Thu 16.11.17**	**Wed 29.11.17**	Work							
121			Project Process Documentation	2 days	Thu 16.11.17	Fri 17.11.17	Work							
122			Tutorial Creation	5 days	Mon 20.11.17	Fri 24.11.17	Work							
123			Release	0 days	Wed 29.11.17	Wed 29.11.17	Work							

Looking4Concerts Page 15

7.2 Future Milestones

1. Milestone 3
   (a) Change password for user (forgot password?)
   (b) Comment categories
   (c) Upvote/downvote comments
   (d) Report users
   (e) Report false concert information
   (f) Upvote reports
   (g) Concert merge
   (h) Improve Concert recommendations
   (i) Improve Concert search
   (j) User search
   (k) Styling improvements, especially in Android
   (l) Populate the database by fetching data from Biletix (optional)
   (m) Write unittests
   (n) Improve the API documentation

8 Code Structure and How We Use Git

Different parts of the whole project stays in the repository separately. Each team works on their own folder, e.g. frontend developers work on lfc-frontend folder, backend developers on lfc-api.

We are developing every new feature in a new branch. When the time comes for this feature to be integrated with the whole project, the developer sends a pull request to his teammates (backend, frontend or Android). They evaluate, then either approve or deny the request, providing an explanation (code review). A pull request cannot be merged to the master branch if it is not approved by at least one team member.

If the request is denied, the author applies the requested changes and notifies the reviewers. The reviewers might request more changes or approve the pull request.

9 Evaluation

9.1 Tools

For back-end, the tools and application programming interfaces (APIs) we used are:

- **Django**: The high-level Python Web framework we used to build the backend.
- **djangorestframework**: The main framework we use on top of the django in implementing back end. We used the serialization part of the rest framework to return data of objects in JSON format in the back-end endpoints.
- **Atom Editor**: is a text editor that's modern and easy to use.
- **simplejwt**: Simple JWT provides a JSON Web Token authentication backend for the Django REST Framework.
- **virtualenv**: A tool to create isolated Python environments. Virtualenv eliminates problems related to versions and dependencies, with a clean pip install from requirements.txt
- **iTerm2**: A Terminal replacement for MacOS. It is much more convenient to use.

- **Sourcetree**: A GUI platform for GitHub. Makes the committing and branching processes very easy!
- **Postman**: A GUI platform for API development. We used it to construct and cache our requests.
- **Spotify Web API**: The official REST API of Spotify which we used to fetch both public (for artist search) and user specific data (for recommendations).
- **Spotipy**: A lightweight Python library for the Spotify Web API. With Spotipy we were able to get full access to all of the music data provided by the Spotify platform, and to user profiles given that we have a valid access token.
- **Wikidata API**: The API of Wikidata, a knowledge base storing structured data. We used it for semantic tags.

For front-end, the tools we used are:

- **React**: Javascript library we used for frontend.
- **HTML**: Markup Language we used to create the website.
- **CSS**: Style Sheet language we used for design.
- **Semantic-ui**: UI framework we used for front end.
- **Google Maps API**: Google's maps API which we used for selecting and displaying concert locations.

For Android, the tools we used are:

- **Android Studio**: the IDE we used to develop our Android app.
- **Retrofit**: A type-safe HTTP client for Android and Java. We used it to write easier request methods.
- **Google Gson Library**: A Java serialization/deserialization library to convert Java Objects into JSON and back. We used it to serialize and deserialize our dto objects.
- **Google Places API**: An API to search for places and get their coordinates, address, etc. provided by Google.
- **Picasso Library**: An image downloading and caching library for Android. We use it to show images using their url's

For project planning, the tool we used is:

- **MS Project**: A project management software, developed by Microsoft.

For communication, the tools we used are:

- **WhatsApp**: The mainstream messaging app. Has faster response time than Slack on average. We discussed our meeting times and places on WhatsApp.
- **Slack**: The popular communication and planning platform for teams which also supports code embeds. We have a channel in sync with GitHub so we get a notification for any activity on issues or commits.
- **GitHub**: A repository hosting service with a Web UI that expedites and complements project management by its issue tracking system, add-commit-push workflow and branching logic for code integration. We especially benefited from the issue system and code reviews.

9.2 Project Management

- The fact that we had an intermediate milestone (16/11/2017) was beneficial.
- Communication can still be improved, but better than before as we added Slack to our communication channels.
- We saw that there can be huge differences in terms of deliverables across teams. For instance, since the front-end team could not catch up, we could not show some features such as; connect to Spotify, simple/advanced search and recommendations.
- On average everyone worked harder than they did for the first milestone.

References

[1] https://github.com/google/gson

[2] http://square.github.io/retrofit/

[3] https://www.djangoproject.com/

[4] https://developers.google.com/places/

[5] http://square.github.io/picasso/

[6] https://github.com/davesque/django-rest-framework-simplejwt

[7] https://spotipy.readthedocs.io/en/latest/