BLG411E Software Engineering

Project Plan

Socialendar

13.10.2017

Socialendar team

Yunus Güngör

Sinan Kartal

Mustafa Sağlam

Mehmet Enes Kayılıoğlu

Recep Can Babaoğlu

1.Introduction

1.1 Scope

The Socialendar application can bring together people who will same events efficiently, easily and precisely. These people are using mobile phone calendar and this app brings together these people in a chatroom if they want. User only need to login to add a new event or to edit current events. Then application take users in a chat room and provide users a communication channel. Users can sign up and sign in with Facebook, Google and mail accounts and application will take their events from device calendar or Facebook. The Socialendar is a JavaScript based application that uses React-Native for mobile phones which have iOS and Android. The application will interface with and utilize third party resources to facilitate all the users' meeting scheduling needs (e.g. database services, email communication, etc.).

1.2 Deliverables

	Deliverables
1	Login to app with Facebook
2	Login to app with Google
3	Login to app with Email
4	Device calendar and cloud database synchronization
5	Assigning chat groups to user events and chatting with users who has same events on their app
6	Showing next free time of the user
7	Ability to add events
8	Ability to view events
9	Ability to view chat
10	Showing weather status for event time and location
11	Adding an event through app to database and device calendar
12	Facebook calendar sync, device calendar and cloud database synchronization

1.3 Epics

No of Epic	Epic	Explanation
1	UI	All of user interface, and interactions with any visible item on screen
2	Database	Connections to Firebase Database (explained in resources), synchronization between device calendar, facebook events and online database
3	Device Calendar	Accessing to device calendar, adding and editing events on device calendar
4	Chatting	Chatting between users
5	Event Matching	Matching user's events on online database with other users' events on online database.
6	Users and login systems	Logging in, signing up, logging out and signing up with other social accounts like Facebook and Google
7	General Structure	General structure based on react-native framework, libraries and react-native modules

1.4 Non-functional Issues

Number of issue	Non-functional Issue	Explanation
1	Usability	Usability will be a big problem in this project since our team doesn't have UI or UX designer or UI guides.
2	Availability	Availability problem mostly solved by Firebase services. Firebase has a very high accessibility percentage for their services.
3	Privacy	Using Firebase database, privacy issues mostly solved. But our code's database accessing functions or database rules can have a vulnerability since our team doesn't have member whom deals with security issues or a penetration tester.
4	Scalability	Scalability is not a serious issue since we use a third party scalable service.
5	Performance	Since JavaScript has a multithreaded nature, performance issues will not be a big problem for our team.

2. Project Plan

Socialander Work Breakdown Structure					
Number of work package	Task	Assigned to	Epic	Dependence to work package	Size
1	Firebase, React native setup	Member 1	7		S
2	Adding necessary modules and libraries	Member 1	7	1	S
3	Authentication	Member 1	6	2	M
4	User log in and log out	Member 1	6	2	M
5	Sign in with Facebook	Member 1	6	3	M
6	Sign in with Google	Member 1	6	3	M
7	Sign in with Mail	Member 1	6	3	M
8	Adding new user on database	Member 1	6	2	M
9	Checking authentication status	Member 1	6	3	M
10	Getting event information from device calendar	Member 2	3	2	M
11	Getting access to Facebook events of user	Member 2	3	2	M
12	Getting data from Facebook events	Member 2	3	11	M
13	Parsing Facebook event data and synchronization	Member 2	2	12	M
14	Synchronization between calendar and online database	Member 2	2	10	M
15	Matching events that occur at the same time on cloud	Member 3	5	14	M
16	Calculating a point between events occur at the same time	Member 3	5	15	M
17	Matching events that has close points	Member 3	5	16	T
18	Creating chat structure	Member 4	4	3	L
19	Being able to send message	Member 4	4	18	M
20	Being able to receive message	Member 4	4	19	M
21	Being able to create chatrooms between users	Member 4	4	20	M
22	Being able to add more than one user to chatrooms	Member 4	4	21	M

23	Creating chatrooms for matched events	Member 3	5	22,17	S
24	Adding events to calendar	Member 2	3	2	M
25	Adding events to database	Member 2	3	2	M
26	Creating add event ui	Member 5	1	24	S
27	Organizing and polishing add event ui	Member 5	1	24	L
28	Creating ui to view events	Member 5	1	14	S
29	Organizing and polishing view event ui	Member 5	1	14	L
30	Creating home screen ui	Member 5	1		M
31	Creating a function to access recent messages	Member 4	4	22	M
32	Showing recent messages on home screen	Member 4	1	22	S
33	Showing events on home screen	Member 3	1	14	M
34	Creating a function to calculate next empty time of the user	Member 2	3	14	M
35	Showing next empty time on home screen	Member 3	1	24	S
36	Creating chat ui	Member 5	1	20	M
37	Organizing and polishing style of chat ui	Member 5	1	20	M
38	Being able to see weather forecast on event time	Member 5	1	29	M
39	Optimizing event points	Member 3	5	17	L
40	Coding event editing function on calendar	Member 3	3	10	M
41	Coding event editing function on database	Member 4	2	10	M
42	Creating event editing ui	Member 4	1	41,40	M

3.Estimates

Number of work package	Man.Week value
1	0.5
2	0.5
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	0.3
18	1.5
19	1
20	1
21	1

Number of work package	Man.Week value
22	1
23	0.5
24	1
25	1
26	0.5
27	1.5
28	0.5
29	1.5
30	1
31	1
32	0.5
33	1
34	1
35	0.6
36	1
37	1
38	1
39	1.5
40	1
41	1
42	1

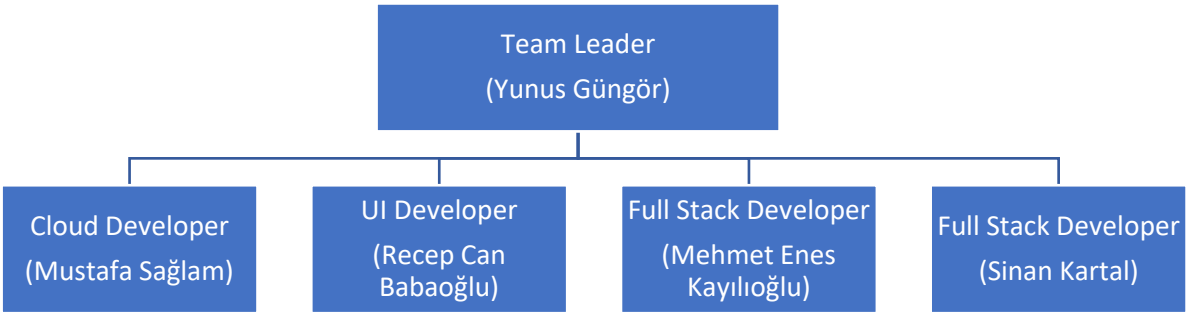
- Estimations calculated by Team Leader based on past experiences on mobile development.

4.Resources

4.1 People

Team Structure

Member no	Name Surname	Roles
Member 1	Yunus Güngör	Team Leader – Backend Developer
Member 2	Sinan Kartal	Full Stack Developer
Member 3	Mustafa Sağlam	Cloud Developer
Member 4	Mehmet Enes Kayılıoğlu	Full Stack Developer
Member 5	Recep Can Babaoğlu	UI Developer

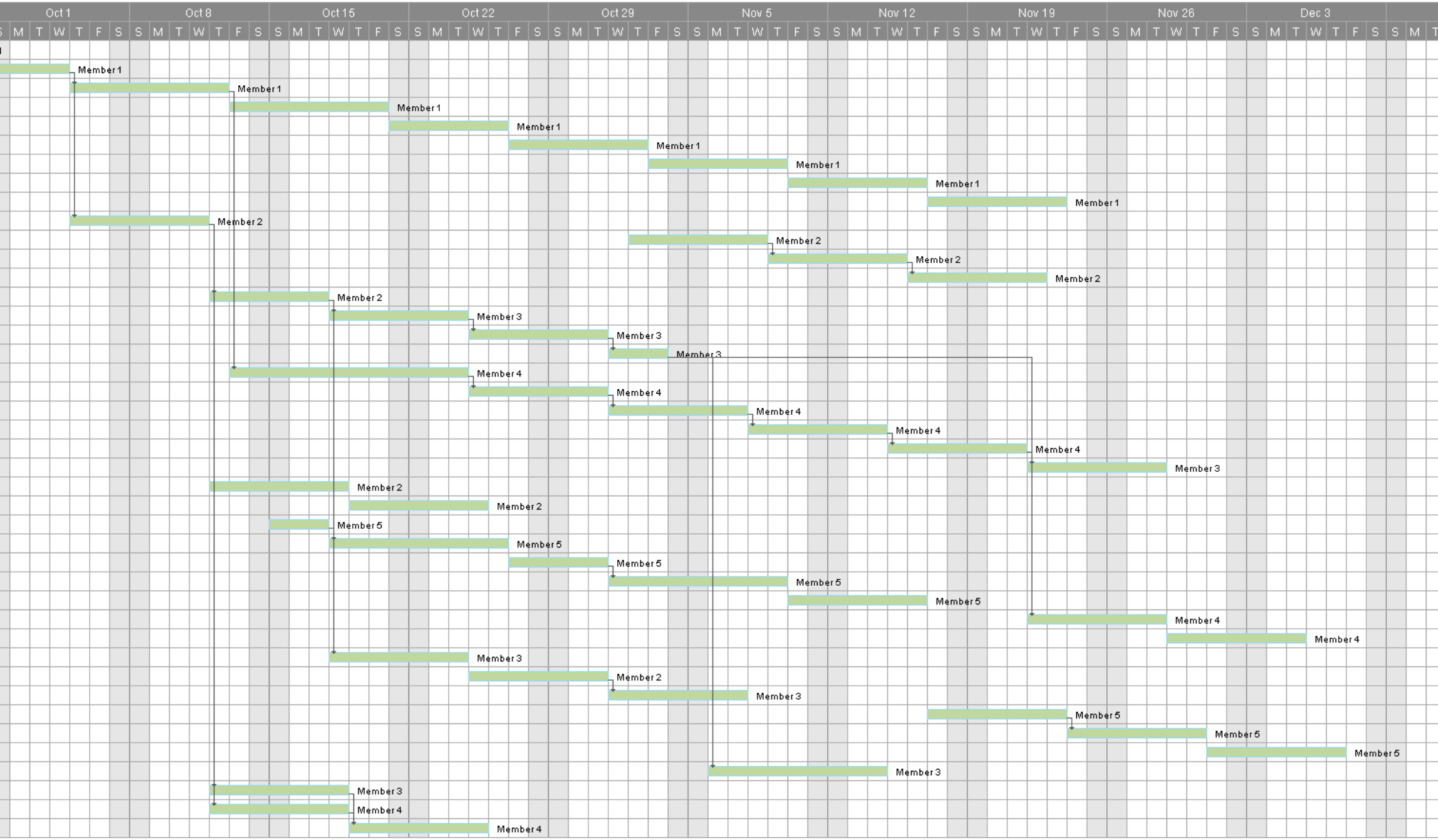


- Team structure is flexible and tasks given in work breakdown structure can be assigned to other member without any problems.

4.2 Online Resources

- Database, authentication and cloud servers: For this service, an all in one tool called Firebase will be used in the project. Firebase is an application development platform that provides API's and online services which is easy to use. Minimal usage until a certain amount has no cost. More information can be found on https://firebase.google.com .

5.Schedule


Oct 1
Oct 8
Oct 15
Oct 22
Oct 29
Nov 5
Nov 12
Nov 19
Nov 26
Dec 3
Member 1
Member 2
Member 3
Member 4
Member 5

6.Risks

	Risk	Probabiltiy	Risk Impact
1.	Wrong time estimation	High	Tolerable
2.	Market development	Low	Serious
3.	Government rule changes	Low	Serious
4.	Privacy Issues	Moderate	Serious
5.	Security Issues	High	Serious
6.	Low communication in team	Moderate	Tolerable
7.	Technological developments	Moderate	Insignificant
8.	Project scope expansions	Low	Tolerable
9.	Steps are not tracked properly	Low	Serious
10.	Wrong budget estimation	Low	Tolerable

	Risk	Explanation
1.	Wrong time estimation	Faulty time estimation
2.	Market development	Changes in aimed market. For example: another application that functions in a similar way or changes in smart phone industry
3.	Government rule changes	Changes in laws that focuses on getting user data (user events etc.), or using that data.
4.	Privacy Issues	Problems with current laws, with users or with other companies on getting user data (user events etc.) or using that data
5.	Security Issues	Losing data, losing integrity of data or losing data protection. This risk especially has a high probability because our team don’t have security advisors or penetration testers
6.	Low communication in team	Communication and trust issues between team members might interfere with app development.
7.	Technological developments	Changes in used libraries and modules or changes in Firebase service might force us to review written code
8.	Project scope expansions	Pivoting app or a part of app or adding new features to app, according to client or users’ requests
9.	Steps are not tracked properly	Missing deadlines, and staying behind schedule
10.	Wrong budget estimation	Estimating the budget with an error