Betitarev

Seroussi Yishay [✡] Bismuth Samuel [❄]

Shaag Yehonatan [✦]

Software Engineering
Assignment 1

November 8, 2018

Abstract

Everyone has already heard about betting. A lot of big companies have exploited all the resources (sports, horses...), using fictive or real money. The problem is, each bet is chosen by the company itself. What if someone want to bet about the number of windows there are in the room, or if he will success the course Software Engineering with grade of at least 85 ?

Given two bettors (at least), and one judge, our application is a platform in which all the bet will run out. Let call Tic the first bettor and Tac the second bettor. Tic is sure than the number of stars in the American flag is 42 and Tac thinks that there are 50 stars. Let Toc be the judge of the bet. Assuming that Tic, Tac, and Toc are already registered in our database, either Tic or Tac must create a new match with the exact bet. Toc designates the winner of the bet.

The general idea is to put what we call a "bet friend" into an application.

[✡] Student of Computer Science (third year), Ariel University, Ariel 40700, Israel. Id : 305027948. Email: seroussi1@gmail.com

[❄] Student of Computer Science (third year), Ariel University, Ariel 40700, Israel. Id : 342533064. Email: samuelbismuth101@gmail.com

[✦] Student of Computer Science (third year), Ariel University, Ariel 40700, Israel. Id : 308357953. Email: yoshago@gmail.com

1 Requirements

The next table 1 is composed of the requirements.

ID	Description	Source	Type	Sub-type	P	Remarks
1	The user must be registered to use the application.	Client	Functional	Operational	5	Two hours of programming. Verification: each mail must appears only once, each user-name only once.
2	The application allows the user to visit profiles of other users, such that he can see statistics about the previous bets of another user.	Client	Functional	Operational	4	Three hours to back end programming and one hour of front end programming.
3	The application allows the user to manage a list of friends.	Users	Functional	Data + Operational	3	One hour of programming. Verification: DO NOT allow double add/remove.
4	The application allows chat between friends	Users	Functional	Operational	2	Eight hours of programming. Verification: There is no possibility to chat between users which are not friends.

5	Each new event (such as chat/bet/friend's request/confirm) must raise a notification by the application in the phone of the relevant user (e.g. in the case of chat's event it will be the friend user who got the message).	Client	Functional	Operational	3	One hour of back end programming and two hours of front end programming. Verification: Viewed messages must be deleted after one day.
6	The application allows two users to bet and, if necessary, an arbitrator can judge and declare the winner (the arbitrator is chosen before the beginning of the bet).	Client	Functional	Operational	5	one hour of back end and three hour of front end programming. Verification: Allow only TWO users and one arbitrator.

7	If there is an arbitrator only he can declare the winner (or he declares that there is a draw). If there is no arbitrator, one user must declare himself as a winner, and the other as a loser, or both of them as even.	Client	Functional	Operational	5	One hour of back end programming, two hours of front end programming.
8	If there is no agreement about the bet, like the gain, or the arbitrator, or even the formulation of the bet, the bet cannot begin.	Client	Functional	Operational	5	One hour of programming.
9	The application allows the next types of gain: Real money or fictive money (via the application), or anything else (text box) (via users).	Client	Functional	Operational	5	Twelve hours of back end programming. Verification: security of the real money transfer.
10	The application doesn't allow to bet real money, unless there is an arbitrator.	Client	Functional	Operational	4	One hour of programming.

11	The application allows to mark a user as a cheater. Also, allows the client to banish any user.	Client	Functional	Operational	2	One hour of programming. Verification: if a user is declared as a cheater more than five times, make sure the client is notified.
12	The system will be available 24/7.	Client	Non functional	QA-Availability	5	Price: 2500 NIS (generator), plus electricity consumption.
13	The user must get a notification of a new event in less than ten seconds (via a phone notification).	Client	Non functional	Performance	3	
14	The application will be usable from February 2019.	Client	Non functional	Management constraint	5	Beta version enter in the market in January 2019.
15	The application use Paypal for money transaction.	Developers	Non functional	QA-Security	5	Price: 400 NIS for a month adding 2 percent of the transaction.
16	The application must be able to be upgraded at any moment by any developer, such that he could add any new feature.	Developers	Non functional	QA-Maintainability	4	Function and classes must be the most clear as possible.

17	The application includes a help library which explains all the features by interactive help.	Users	Functional	Operational	1	One hour of programming.
18	The server shall support at least 1000 users at the same time, and at least 50 Terabyte.	Developers	Non functional	Hardware constraint	3	Adding servers if needed.
19	The application saves all the data of all the users.	Client	Functional	Data	2	One hour of programming.

Note 1: *The column P means priority which is a grade given between 1 and 5, where 5 is a mandatory requirement, and 1 is a least important requirement.*
Note 2: *When we use the word "user(s)" in the source column, we mean survey among friends.*

Table 2: Requirements