Overview:

The application is designed to help football enthusiasts keep track of their favorite team. Designed for fans with busy schedules; the service enables one to keep up with the latest happenings at their preferred football club. The recurring messages will be channeled through Twilio and the backend will pull data from Twitter to get live updates. User preferences and Schedule of matches will be stored in a database. Ideally, weekly updates on upcoming matches will be provided along with kickoff timings. On match day, the user can decide the level of information he/she prefers choosing from basic updates to possible live commentary updates. They will also have an option of stopping the live updates at any point as per their discretion. The service will possibly provide highlights of the match at the end, along with league table standings. The app will be designed for minimal user effort, prioritizing usability. No pulling up information from the internet again.

Functions:

- **Setting up user preferences:** User can choose multiple clubs to follow from wide variety of top European clubs
- **Existing user:** Welcomes user , lists previously chosen preferences
- **Modify/ Delete Preference**: Based on need user can add or delete existing football club preference
- **Live commentary update:** User can choose to get play-by-play update from the match
- **League table standings:** User will get updated league table standings
- **Video highlights:** User will get highlights of goals, highlights if this feature is enabled
- **Stopping live updates:** User will be able to turn off all updates at will as per need for any game
- **Help page:** Lists keywords for enabling various functions
- **Bot limitation error:** For unknown keywords an error will show up to enhance usability
- Top tweets from club channel: User will be able to get updates on major club news sourced from the official handle

User need/ Effort	Low	Medium	High
Low			Help page, Error handling
Medium	League table standings	Modify/Delete Preference, Commentary Update, Highlights	
High		Existing user, Twitter	Onboarding

Under progress
Completed

Figure.1 Functionality Matrix

Workflow: The bot is designed to minimize the user effort. This can be seen in the workflow structure shown below (figure 2). The onboarding process is done once and it enables the user to set up his/her

preferences. The service allows selection of multiple teams across the 4 major football leagues across the world. The user team preferences, along with basic details are stored in a database. Next the user is asked to select recurring services such as upcoming matches, league table and live updates. In the future, the user will have the option of commencing or terminating live commentary updates on-demand. The user involvement is top-heavy, a one-time registration setting up majority of the services offered by FootBot.
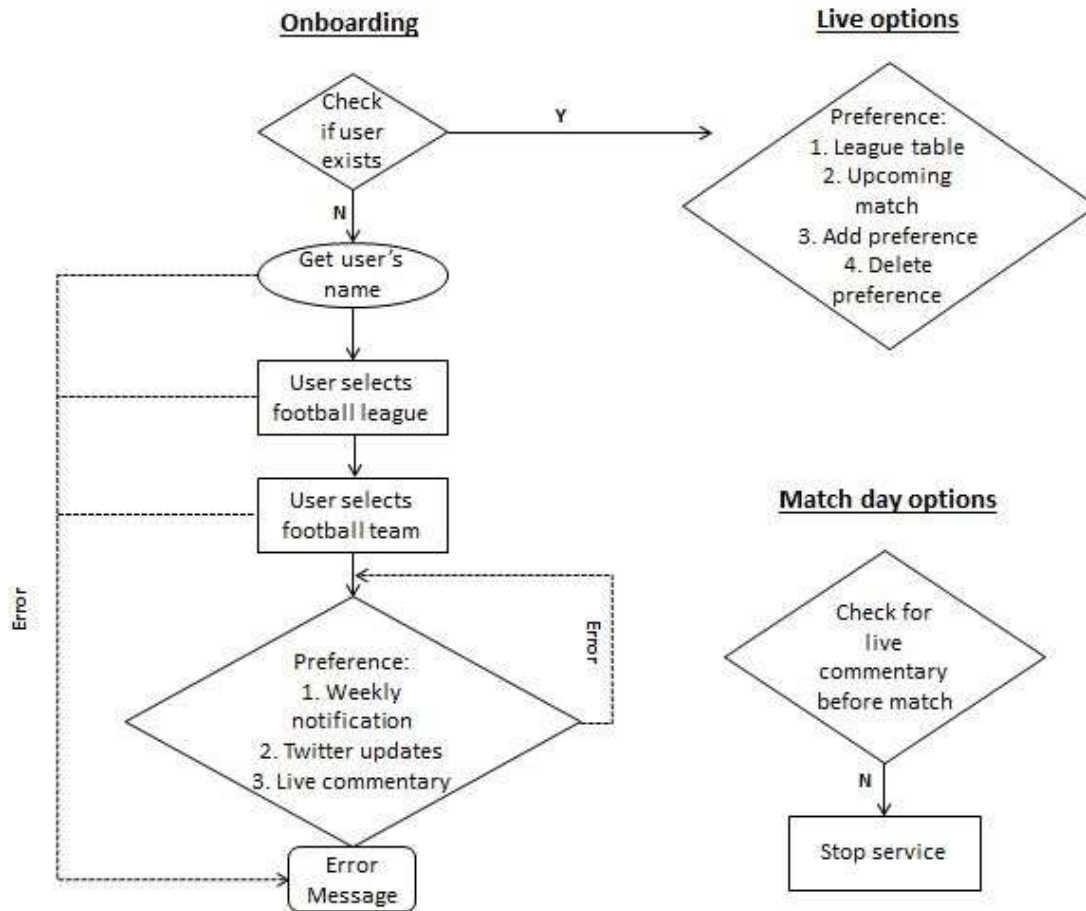
Onboarding

Figure.2 Workflow diagram

Bot personality:
Andy is the embodiment of FootBot. He is a gruff, retired West Ham hooligan who breathes football. You would find him watching all matches in a pub with his mates. The bot's personality is heavily drawn from Andy's and is exhibited in the language and tone of the interaction. The idea was to find the balance between being excessive in humanizing the experience and being overbearing.

Backend implementation: The application uses "Twilio" API for SMS based interactions and data from the "Football data" API using the httparty gem. Team and league specific numbers for specific teams are stored in a database called *Teamdetails*. *User* and *Preference* database table are used to store user details and preferences.
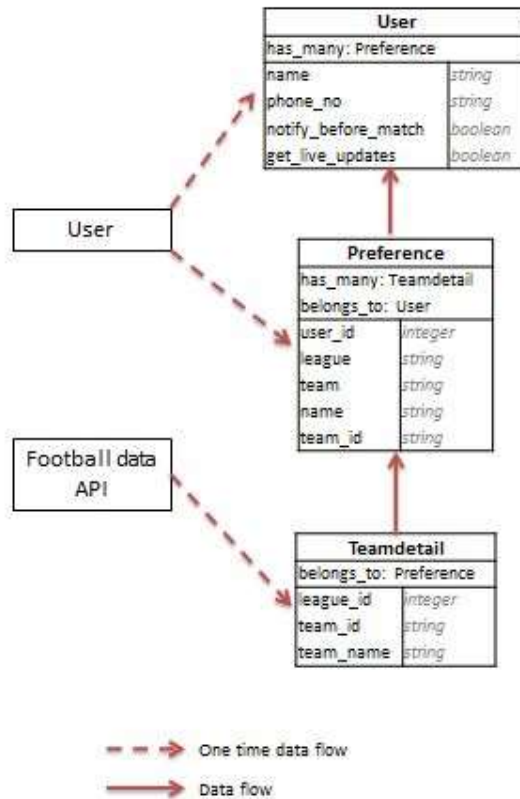

Figure3. Data flow diagram

Current Status:

The bot is currently still in the development stage and needs to be polished before it can be deployed publically. The program works successfully in pulling data from the Football Data API, in addition to working with the Twilio API. Screenshots indicating the interaction including the end points are displayed below (figure 4). The onboarding and Twitter features need tweaking before they can function independently.
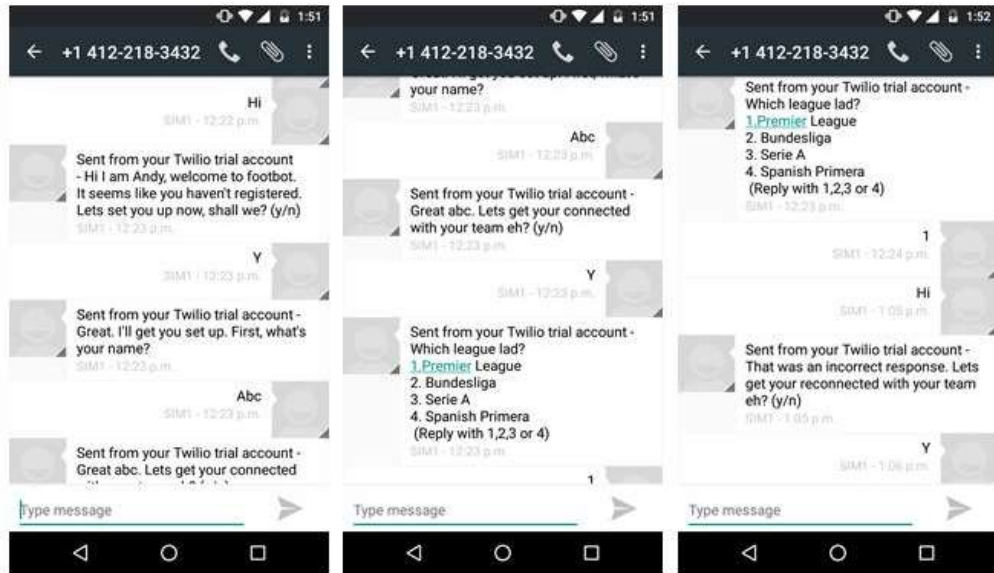
Figure 4. Interaction screenshot

Figure 3.Error handling

Reflection:

The bot showcases the progress made through the course of the online prototyping class. Despite limited functionality, the bot successfully manages to integrate different API's under one application. Moving forward, I hope to work on improving the interaction, experience and breadth of the functionality offered by FootBot