

POLITECNICO DI MILANO



POLITECNICO
MILANO 1863

DESIGN AND IMPLEMENTATION OF MOBILE
APPLICATIONS

iSport

Design Document

di
Gianluigi Oliva

January 26, 2019

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Intended Audience	3
1.3	Definitions, acronyms, abbreviations	3
1.4	Mobile Application Scope	4
1.5	Framework	5
1.6	Functional Requirements	5
1.7	Non Functional Requirements	6
1.8	Assumptions, Dependencies and Constraints	6
2	Architecture	7
2.1	Database	7
2.2	Client	7
3	Use case functional requirements analysis	8
4	Sequence Diagram	20
5	User Interfaces	27
6	External Services and Libraries	28
7	Software System Attribute	32
7.1	Reliability	32
7.2	Availability	32
7.3	Security	32
7.4	Maintainability	32
7.5	Usability	32
8	Test Cases	33
9	Cost Estimation	34

1 Introduction

1.1 Purpose

Questo documento descrive le fasi di progettazione e prototipazione per la realizzazione dell'applicazione mobile "iSport". In dettaglio, verranno discussi i componenti principali, le funzionalità e l'esperienza dell'utente.

iSport è una applicazione in cui obiettivo principale è la visualizzazione di informazioni e dati relativi al mondo dello sport. In particolare ci si focalizzerà sulle notizie giornalistiche più rilevanti e sui dati relativi alle partite di calcio della giornata.

Questo progetto è il risultato dell'implementazione delle conoscenze acquisite durante il corso "Design and Implementation of Mobile Applications" fornito dal Politecnico di Milano.

1.2 Intended Audience

This document is produced for those who develop, evaluate and use iSport mobile application:

- The engineers who had the idea and developed the application.
- The testers that must verify the effective implementation of all the described components and functions.
- The user who will use the application and take advantage of its functionalities.
- The future contributors who wish to develop new features.

1.3 Definitions, acronyms, abbreviations

Definitions

- **Platform:** The application as a whole.
- **User:** An end user who will use the application
- **Match:** A match between two teams that has already occurred or is in progress

- **Framework:** Reusable set of libraries or classes for a software system.
- **News:** Una notizia relativa al mondo dello sport presente su qualche rivista giornalistica
- **Pronostico:** Una previsione sul risultato di un match appartenente ad una classe di previsioni possibili
- **Quota:** Il valore di retribuzione di un pronostico relativo ad un determinato match
- **REST:** is a way of providing interoperability between computer systems on the Internet.

Acronyms

- **MVC:** Model - View - Controller
- **HTTPS:** HyperText Transfer Protocol Secure
- **IDE:** Integrated Development Environment
- **API:** Application Programming Interface
- **JSON:** JavaScript Object Notation
- **UML:** Unified Modelling Language.
- **UX:** User Experience
- **URL:** Uniform Resource Locator

Abbreviations

- **App:** Mobile Application

1.4 Mobile Application Scope

iSport è stato sviluppato per tutti coloro che sono appassionati di sport cercando di unificare sotto un'unica applicazione tutti i servizi presenti sul mercato. In questo modo si vuole dare più continuità di utilizzo all'utente finale, senza che egli abbia la necessità di navigare su più applicazioni per ottenere lo stesso risultato.

In particolare l'applicazione si articolerà in tre schermate:

- **News**
- **Live**
- **Bet**

Nella sezione "News" saranno presenti le principali notizie relative al mondo dello sport visualizzabili con un'immagine di anteprima e una piccola descrizione. Inoltre premendo sulla singola notizia sarà possibile visualizzare l'articolo completo.

Nella sezione "Live" saranno presenti tutte le partite della giornata corrente con il risultato della partita se già conclusa o quello attuale se ancora in corso. Premendo sulla singola partita sarà possibile consultare tutte le informazioni su di essa come i marcatori e il minuto del goal, cartellini, formazione e statistiche.

Nella sezione "Bet" saranno presenti le quote relative ai principali pronostici delle partite odierne. Premendo sulle singole quote sarà possibile comporre una schedina e una volta conclusa l'applicazione calcolerà la potenziale vincita in base all'importo della giocata.

1.5 Framework

Lo sviluppo di iSport è stato realizzato mediante l'utilizzo delle native SDK iOS, in particolare facendo ricorso al linguaggio di programmazione Swift. Questa scelta è stata fatta per permettere un maggior controllo delle risorse del sistema e l'accesso a dei servizi di sistema non possibili con l'utilizzo di framework cross-platform come PhoneGap o React Native. I principali obiettivi sono stati l'implementazione di diverse funzionalità e l'integrazione con altri siti.

1.6 Functional Requirements

The product provide to users a simple and user-friendly interface to:

- Visualizzare le anteprime delle notizie
- Visualizzare le notizie complete
- Visualizzare i risultati delle partite in tempo reale
- Visualizzare i marcatori delle singole partite
- Visualizzare le ammonizioni delle singole partite
- Visualizzare le formazioni delle due squadre di una partita
- Visualizzare le statistiche delle singole partite
- Visualizzare le quote dei principali pronostici delle singole partite
- Comporre la propria schedina
- Visualizzare la potenziale vincita della schedina

1.7 Non Functional Requirements

The application must be able to:

- Run both on phone and tablet (only if they have an iOS).
- Funzionare senza richiedere permessi a dati sensibili dell'utente e servizi che potrebbero necessitare un costo dell'utente (come chiamate o SMS).
- occupare tutto lo schermo a disposizione.
- l'applicazione deve mantenere le preferenze e lo stato ad ogni avvio.

1.8 Assumptions, Dependencies and Constraints

Costraints

- **Hardware limitations:** our application runs on every mobile device like smartphones and tablets. Therefore, as the App consumes a low amount of RAM, the only hardware constraint for the users is to have a mid-range device. (for instance iPhone 5 or better).
- **Parallel operations:** the application must be able to handle multiple parallel requests with high reactivity.

Assumptions and Dependecies

- **Internet Connection:** the device used by the users dispose of an internet connection and a sufficient bandwidth to use the application.
- **No privileged users:** there are no priviled users or administrators with particular functions.
- **No user connections:** every user is independent from the others.
- **API availability:** the API provided by third part's services are always available.
- **OS Permission Granted:** the user will always grant to his OS's device the permission to access to all the needed services.

2 Architecture

2.1 Database

Dal momento che l'applicazione riceve tutti i dati necessari da servizi esterni tramite delle API, gli unici dati che necessitano di un salvataggio sono le partite che costituiscono una schedina. Inoltre non essendo presente un'interazione tra i diversi utenti, i dati vengono salvati in locale sfruttando il Core Data presente nell'SDK di iOS.



2.2 Client

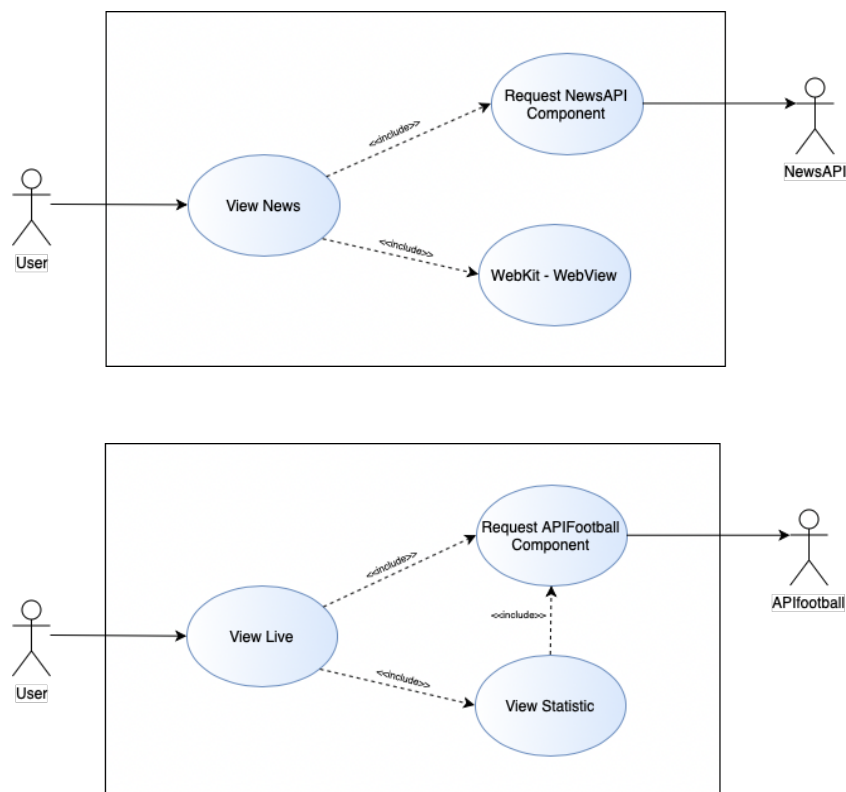
Per la realizzazione dell'applicazione si è optato per un back-end mobile, quindi un'architettura solo client. Questa scelta deriva dal fatto che l'applicazione non ha la necessità di interfacciarsi con altri utenti e per i vari servizi fa uso di API terze. La comunicazione con i servizi terzi si basa su richieste HTTPS REST, in particolare tramite delle richieste GET.

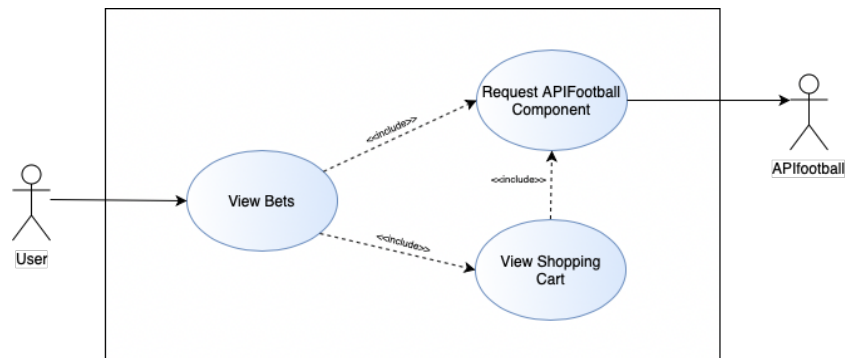
Il client si basa sul tradizionale pattern MVC.

- **Model:** in this package are contained all the classes about data shown to user, taken by the Controller and published by the View.
- **View:** in questo package sono contenuti tutti i componenti responsabili della visualizzazione dei dati e dell'interazione con l'utente
- **Controller:** in questo package sono contenuti tutti gli oggetti responsabili dell'interazione tra uno o più oggetti view dell'applicazione e uno o più oggetti model.

3 Use case functional requirements analysis

This section describes how actors can interact with iSport in order to use all the features implemented in the app. The focus of this part is on the front-end and we show the operations that can be performed by the actors without taking care of the system architecture behind the app.





Visualizzare Notizie

Name	Visualizzare Notizie
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	1, 2
Event Flow	<ul style="list-style-type: none"> • L'utente apre l'applicazione • L'utente preme sulla tab "News" • L'applicazione fornisce un elenco delle notizie principali • L'utente preme sull'articolo di cui vuole visualizzare la notizia completa • L'utente preme il tasto "Done" per terminare la lettura
Exit condition	L'utente ha letto la notizia di interesse.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere gli articoli e quindi non può mostrarle. L'articolo è stato rimosso dal sito di origine, ma non nel database del servizio terzo

Visualizzare Partite

Name	Visualizzare Partite
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	3
Event Flow	<ul style="list-style-type: none">• L'utente apre l'applicazione• L'utente preme sulla tab "Live"• L'applicazione fornisce un elenco delle partite del giorno
Exit condition	L'utente ha visualizzato i risultati delle partite.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Visualizzare Formazione Partita

Name	Visualizzare Formazione Partita
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	3, 6
Event Flow	<ul style="list-style-type: none">• L'utente apre l'applicazione• L'utente preme sulla tab "Live"• L'applicazione fornisce un elenco delle partite del giorno• L'utente preme sulla partita di cui vuole ottenere l'informazione richiesta• L'utente preme sul bottone raffigurante il campo di gioco
Exit condition	L'utente ha visualizzato le formazioni delle due squadre della partita richiesta.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Visualizzare Marcatori Partita

Name	Visualizzare Marcatori Partita
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	3, 4
Event Flow	<ul style="list-style-type: none">• L'utente apre l'applicazione• L'utente preme sulla tab "Live"• L'applicazione fornisce un elenco delle partite del giorno• L'utente preme sulla partita di cui vuole ottenere l'informazione richiesta• L'utente preme sul bottone raffigurante un pallone
Exit condition	L'utente ha visualizzato i marcatori della partita richiesta.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Visualizzare Statistiche Partita

Name	Visualizzare Statistiche Partita
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	3, 7
Event Flow	<ul style="list-style-type: none"> • L'utente apre l'applicazione • L'utente preme sulla tab "Live" • L'applicazione fornisce un elenco delle partite del giorno • L'utente preme sulla partita di cui vuole ottenere l'informazione richiesta • L'utente preme sul bottone raffigurante un grafico
Exit condition	L'utente ha visualizzato le statistiche della partita richiesta.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Visualizzare Ammonizioni Partita

Name	Visualizzare Ammonizioni Partita
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	3, 5
Event Flow	<ul style="list-style-type: none"> • L'utente apre l'applicazione • L'utente preme sulla tab "Live" • L'applicazione fornisce un elenco delle partite del giorno • L'utente preme sulla partita di cui vuole ottenere l'informazione richiesta • L'utente preme sul bottone raffigurante un cartellino
Exit condition	L'utente ha visualizzato gli ammoniti ed espulsi della partita richiesta.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Visualizzare Quote

Name	Visualizzare Quote
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	8
Event Flow	<ul style="list-style-type: none">• L'utente apre l'applicazione• L'utente preme sulla tab "Bet"• L'applicazione fornisce un elenco delle quote dei principali pronostici delle partite odierne
Exit condition	L'utente ha visualizzato le quote delle partite odierne.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Visualizzare la Schedina

Name	Aggiungere Partita a Schedina
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	9
Event Flow	<ul style="list-style-type: none">• L'utente apre l'applicazione• L'utente preme sulla tab "Bet"• L'utente preme sul bottone raffigurante un carrello nella NavBar
Exit condition	L'utente visualizza i pronostici presenti nella schedina.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Aggiungere Partita a Schedina

Name	Aggiungere Partita a Schedina
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	9
Event Flow	<ul style="list-style-type: none"> • L'utente apre l'applicazione • L'utente preme sulla tab "Bet" • L'applicazione fornisce un elenco delle quote dei principali pronostici delle partite odierne • L'utente preme sulla quota relativa al pronostico e partita da aggiungere
Exit condition	L'utente ha aggiunto un pronostico di una partita alla schedina.
Exceptions	L'utente non è connesso alla rete e quindi non può inviare le richieste per ottenere i risultati e quindi non può mostrargli. La connessione potrebbe essere debole e quindi non riuscire a ricevere i dati richiesti.

Eliminare Partita da Schedina

Name	Eliminare Partita a Schedina
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	9
Event Flow	<ul style="list-style-type: none"> • L'utente apre l'applicazione • L'utente preme sulla tab "Bet" • L'utente preme sul bottone raffigurante un carrello nella NavBar • L'utente effettua uno swipe verso sinistra sulla partita da eliminare
Exit condition	L'utente ha eliminato un pronostico di una partita alla schedina.
Exceptions	Nessuna Eccezione

Calcolare potenziale Vincita Schedina

Name	Calcolare potenziale Vincita Schedina
Actor	User
Entry Condition	L'attore ha scaricato l'applicazione
Goal	10
Event Flow	<ul style="list-style-type: none">• L'utente apre l'applicazione• L'utente preme sulla tab "Bet"• L'utente preme sul bottone raffigurante un carrello nella NavBar• L'utente inserisce l'importo da giocare nell'apposito input• L'utente preme il tasto "Done"
Exit condition	L'utente visualizza la potenziale vincita della schedina attuale.
Exceptions	Nessuna Eccezione

4 Sequence Diagram

In this part we show the logical flows behind some features in order to explain our work and to facilitate future implementations. The sequence diagrams will describe the interactions between the different parts of the system and the user.

View News

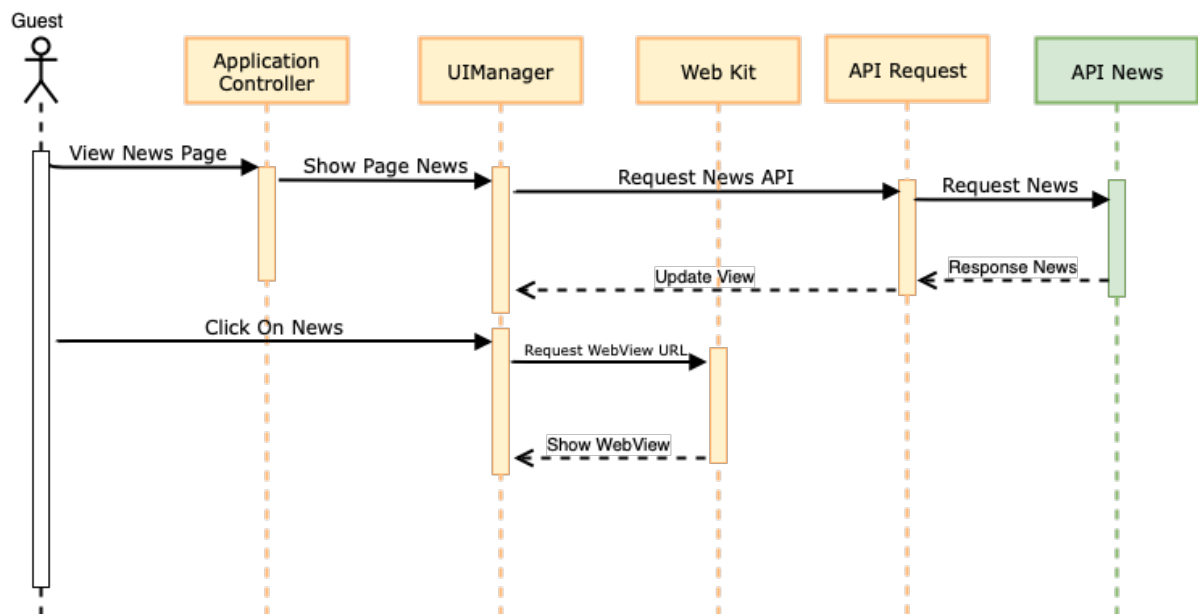
La procedura di "View News" inizia quando l'utente apre l'applicazione o preme sull'oggetto del Tab Bar Navigator relativa alla sezione "News". Il sequence diagram mostra la normale procedura.

Dopo aver attivato la sezione news l'applicazione invierà al servizio di "NewsAPI" una richiesta per ottenere tutte le informazioni relative alle principali notizie del giorno.

Una volta ottenute tali informazioni il Controller si occuperà di creare una UITableViewCell per ogni notizia e di inserirle nella TableView.

In maniera asincrona, inoltre, scaricherà le immagini di anteprima per permettere comunque all'utente di poter leggere i titoli delle notizie.

Quando una cella viene premuta il Controller aprirà una WebView indirizzata all'URL della notizia in modo da permettere all'utente di leggere l'articolo completo.



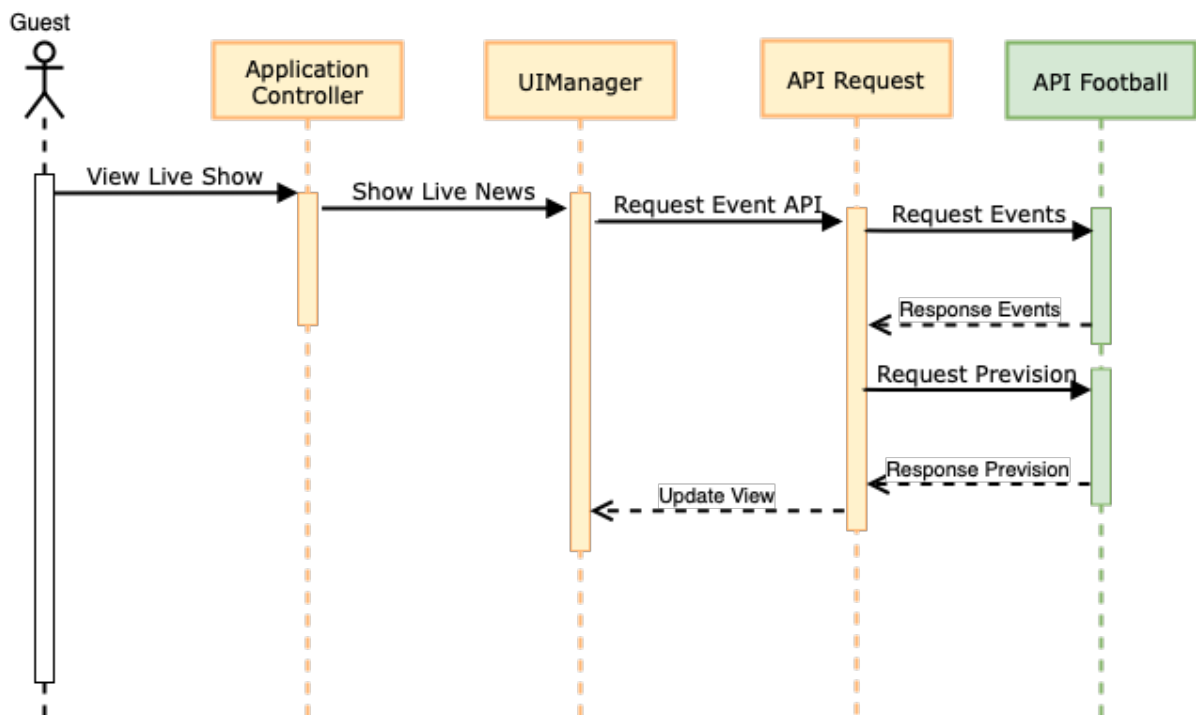
View Match

La procedura di "View Match" inizia quando l'utente preme sull'oggetto del Tab Bar Navigator relativa alla sezione "Live".

Dopo aver attivato la sezione "Live" l'applicazione invierà al servizio di "APIFootball" una richiesta per ottenere le informazioni relative alle partite della giornata e le previsioni sulle stesse.

Una volta ottenute tali informazioni il Controller si occuperà di creare una UITableViewCell per ogni notizia e di inserirle nella TableView.

Periodicamente il Controller rinvierà tali richieste per mantenere lo stato delle partite aggiornate.

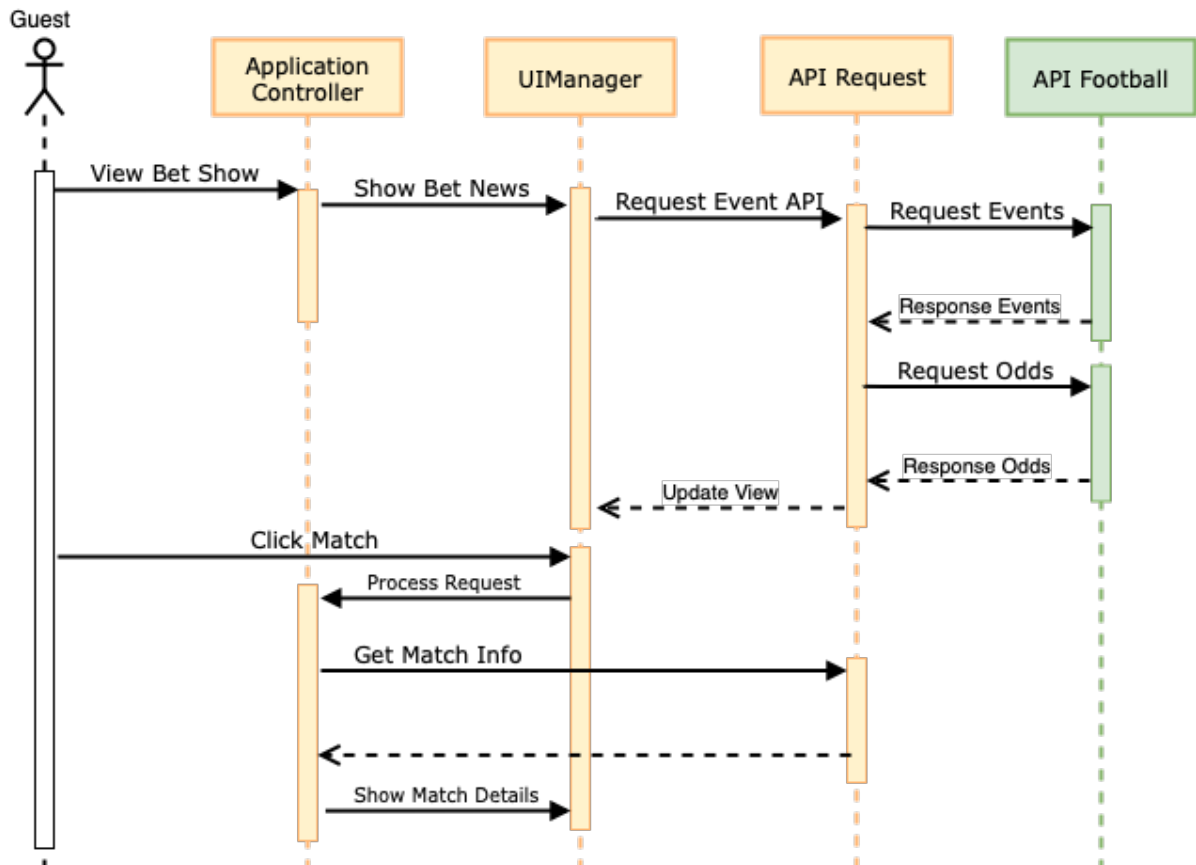


View Match Details

La procedura di "View Match" inizia quando l'utente si trova nella sezione "Live" e preme su un match di cui vuole conoscerne i dettagli.

A quel punto il Controller accede ai dati relativi allo specifico match, scaricati in precedenza dal servizio di "APIFootball".

Successivamente crea delle View per ogni informazione fornita dal servizio. Una volta aperta la pagina di dettaglio l'utente può cambiare l'informazione desiderata tramite i relativi pulsanti.



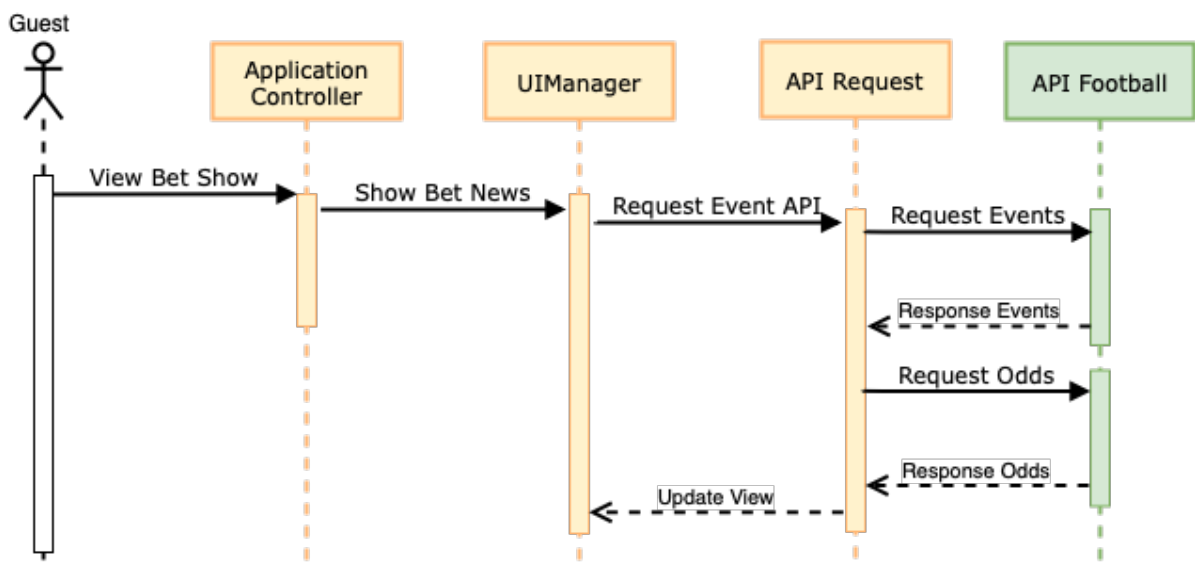
View Odds

La procedura di "View Odds" inizia quando l'utente preme sull'oggetto del Tab Bar Navigator relativa alla sezione "Bet".

Dopo aver attivato la sezione "Bet" l'applicazione invierà al servizio di "APIFootball" una richiesta per ottenere le informazioni relative alle partite della giornata e alle quote sulle stesse.

Una volta ottenute tali informazioni il Controller si occuperà di creare una UITableViewCell per ogni notizia e di inserirle nella TableView. In particolare in ogni cella verranno inseriti tre pulsanti relativi ai pronostici principali con indicate le relative quote.

Periodicamente il Controller rinvierà tali richieste per mantenere lo stato delle partite aggiornate.



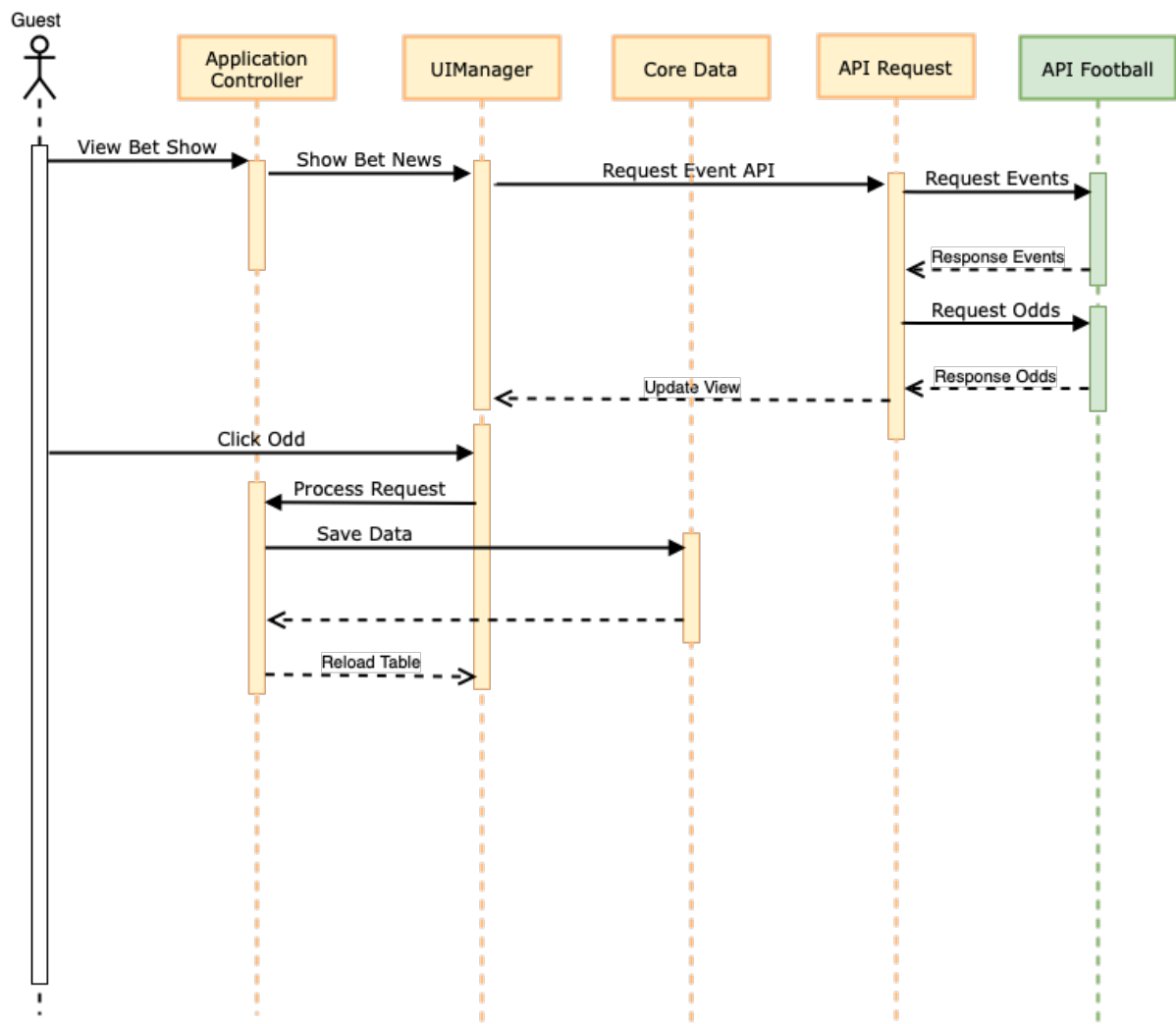
Add Odd

La procedura di "Add Odd" inizia quando l'utente si trova nella sezione "Bet" e preme sul pronostico del match che vuole aggiungere alla schedina.

A quel punto il Controller accede ai dati relativi allo specifico match, scaricati in precedenza dal servizio di "APIFootball", creando la struttura dati da inserire nel Database.

Una volta creata la struttura dati il Controller verificherà se il match è già presente nel database. Se presente aggiornerà il contenuto dell'elemento con il nuovo pronostico, altrimenti creerà un nuovo record.

Terminato l'aggiornamento del DB il Controller aggiornerà la View corrente per mostrare un feedback visivo dell'avvenuta modifica.

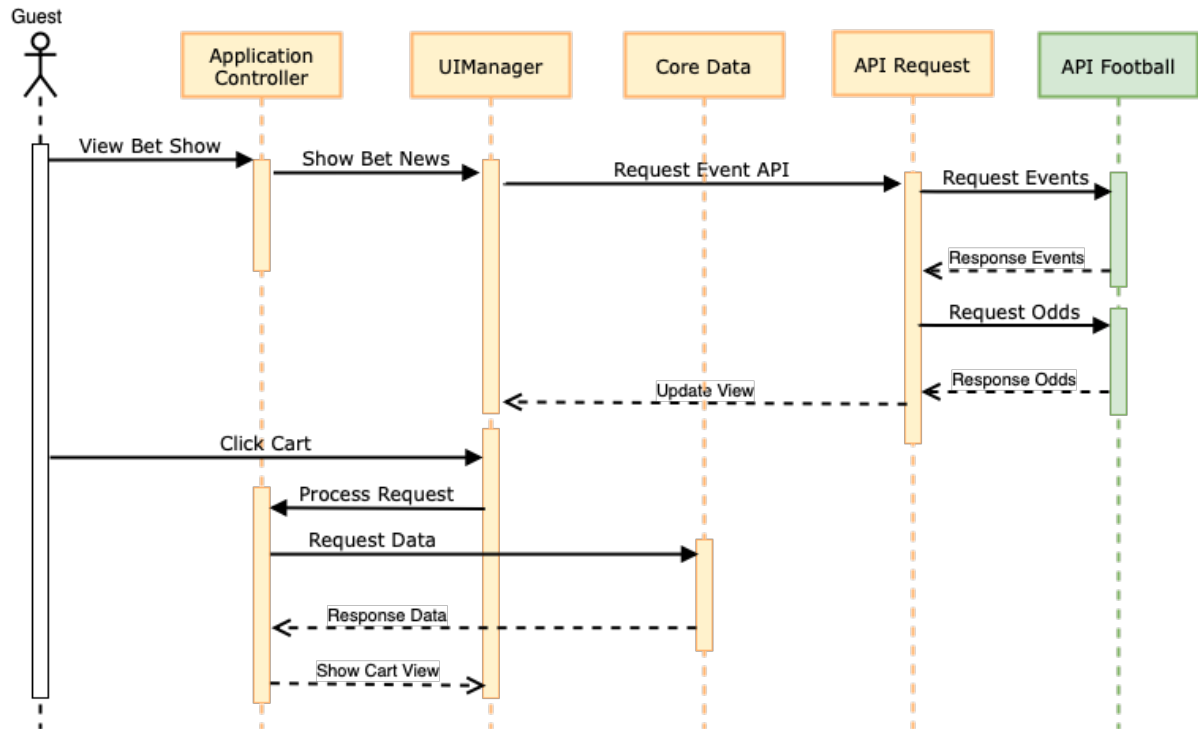


View Cart

La procedura di "View Cart" inizia quando l'utente si trova nella sezione "Bet" e preme sul bottone raffigurante un carrello nella Bar Navigation.

A quel punto il Controller accede ai dati presenti nel Database e per ognuno di essi creerà una UITableViewCell da inserire nella TableView.

L'utente può poi modificare l'importo giocato dall'input corrispondente. Una volta aggiornato l'importo il Controller calcolerà la potenziale vincita andando a moltiplicare tra loro tutte le quote e l'importo giocato.

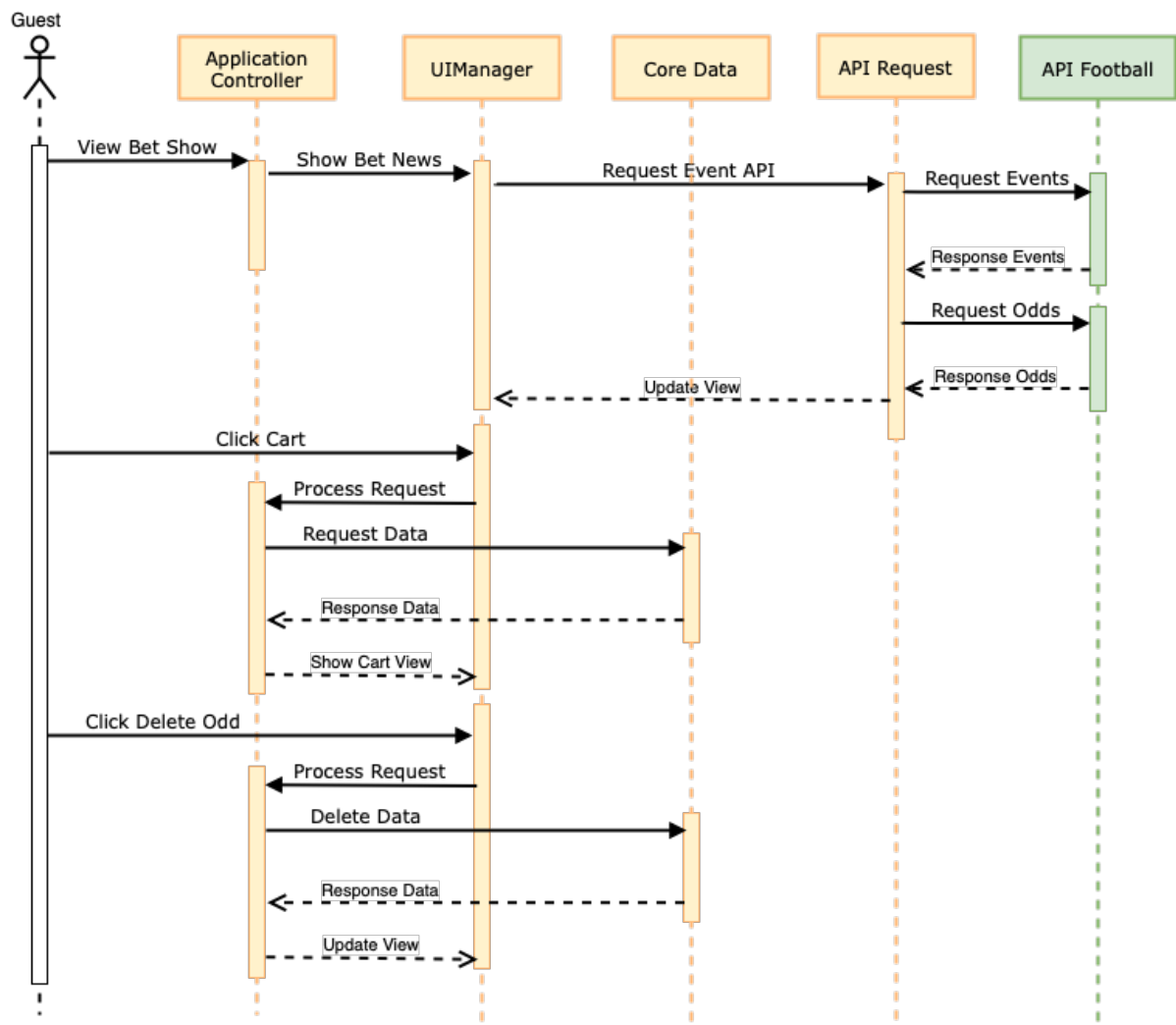


Remove Odd

La procedura di "Remove Odd" inizia quando l'utente si trova nella sezione "Bet" e in particolare una volta visualizzato il carrello attuale.

Una volta visualizzata la lista di tutti i pronostici presenti nel carrello l'utente può eliminarne uno mediante uno swipe verso sinistra sulla cella corrispondente.

A quel punto il Controller eliminerà il corrispondente record dal Database e quindi aggiornerà il contenuto del carrello ricalcolando anche la nuova potenziale vincita.



5 User Interfaces

6 External Services and Libraries

iSport application uses several third parties API in order to provide all the services to the user.

Per il parsing dei dati JSON è stato usato il componente JSONDecoder nativo dell'UIKit.

NewsAPI

Quando un utente vuole visualizzare le principali notizie del giorno l'applicazione esegue le seguenti operazioni:

- Esegue una richiesta GET al l'endpoint fornito dal servizio
- Se la risposta è valida effettua il parsing del contenuto JSON
- Crea le celle della tabella contenente le informazioni fornite dal servizio e parsate
- Scarica in maniera asincrona le immagini di anteprima

L'endpoint utilizzato è "https://newsapi.org/v2/top-headlines?country=it&category=sports&apiKey=API_KEY". La risposta fornita dal servizio è:

```
{
  status: "ok",
  totalResults: 70,
  articles: [
    {
      source: {
        id: null,
        name: "Milannews.it"
      },
      author: "Salvatore Trovato",
      title: "Milan, La Repubblica: \"Vincoli e crescita, Gazidis indica la via di mezzo\" - Milan News",
      description: "Un \"sentiero ragionevole\" lungo il quale possano camminare insieme sostenibilità finanziaria e libertà di investimento. È questa - riporta il quotidiano La Repubblica",
      url: "https://www.milannews.it/news/milan-la-repubblica-vincoli-e-crescita-gazidis-indica-la-via-di-mezzo-321868",
      urlToImage: "https://net-storage.tccstatic.com/storage/milannews.it/img_notizie/thumb3/34/34bd9b6fe7d0e1748194-639c61846db6balfb8fe64bca2ce28f5.jpeg",
      publishedAt: "2019-01-25T09:37:26Z",
      content: "Un \"sentiero ragionevole\" lungo il quale possano camminare insieme sostenibilità finanziaria e libertà di investimento. È questa - riporta il quotidiano La Repubblica - la via di uscita dal braccio di ferro tra Milan e Uefa suggerita da Ivan Gazidis. \"Lo scop... [+300 chars]"
    },
    {
      source: {
        id: null,
        name: "Tuttojuve.com"
      }
    }
  ]
}
```

Per il parsing quindi è stata usata la seguente struttura dati:

```
struct Article: Decodable {
    let author: String?
    let title: String?
    let description: String?
    let url: String?
    let urlToImage: String?
    let publishedAt: String?
}

struct Articoli: Decodable {
    let status: String
    let articles: [Article]
}
```

APIFootball

Quando un utente vuole visualizzare i risultati delle partite del giorno l'applicazione esegue le seguenti operazioni:

- Esegue una richiesta GET al l'endpoint fornito dal servizio
- Se la risposta è valida effettua il parsing del contenuto JSON
- Crea le celle della tabella contenente le informazioni fornite dal servizio e parsate
- Aggiorna le informazioni ogni 30 secondi per mantenere i dati in tempo reale

L'endpoint utilizzato per ottenere i risultati è "https://apifootball.com/api/?action=get_events".
La risposta fornita dal servizio è:

```
[
  {
    "match_id": "277488",
    "country_id": "170",
    "country_name": "Italy",
    "league_id": "79",
    "league_name": "Serie A",
    "match_date": "2018-04-22",
    "match_status": "FT",
    "match_time": "15:00",
    "match_hometeam_name": "Juventus",
    "match_hometeam_score": "0",
    "match_awayteam_name": "SSC Napoli",
    "match_awayteam_score": "1",
    "match_hometeam_halftime_score": "0",
    "match_awayteam_halftime_score": "0",
    "match_hometeam_extra_score": "",
    "match_awayteam_extra_score": "",
    "match_hometeam_penalty_score": "",
    "match_awayteam_penalty_score": "",
    "match_hometeam_system": "4-3-2-1",
    "match_awayteam_system": "4-3-3",
    "match_live": "1",
    "goalscorer": [
      {
        "id": 1,
        "name": "C. Ronaldo",
        "number": 7,
        "goals": 1,
        "order": 1
      }
    ],
    "cards": [
      {
        "id": 1,
        "name": "C. Ronaldo",
        "number": 7,
        "goals": 1,
        "order": 1
      }
    ],
    "lineup": [
      {
        "id": 1,
        "name": "C. Ronaldo",
        "number": 7,
        "goals": 1,
        "order": 1
      }
    ],
    "statistics": [
      {
        "id": 1,
        "name": "C. Ronaldo",
        "number": 7,
        "goals": 1,
        "order": 1
      }
    ]
  }
]
```

6 External Services and Libraries

Per il parsing quindi è stata usata la seguente struttura dati:

```
struct Partita: Decodable {
    let leagueName: String?
    let matchId: String?
    let matchStatus: String?
    let matchTime: String?
    let matchDate: String?
    let matchHometeamName: String?
    let matchHometeamScore: String?
    let matchAwayteamName: String?
    let matchAwayteamScore: String?
    let matchHometeamHalftimeScore: String?
    let matchAwayteamHalftimeScore: String?
    let goalscorer: [Goallist]
    let cards: [CardList]
    let statistics: [Statistic]
    let lineup: Formazione
}

struct Goallist: Decodable {
    let time: String?
    let homeScorer: String?
    let score: String?
    let awayScorer: String?
}

struct CardList: Decodable {
    let time: String?
    let homeFault: String?
    let card: String?
    let awayFault: String?
}

struct Statistic: Decodable {
    let type: String?
    let home: String?
    let away: String?
}

struct Formazione: Decodable {
    let home: Campo?
    let away: Campo?
}

struct Campo: Decodable {
    let startingLineups: [Lineup]?
    let substitutions: [Lineup]?
}

struct Lineup: Decodable{
    let lineupPlayer: String?
    let lineupNumber: String?
    let lineupPosition: String?
    let lineupTime: String?
}
```

Per ottenere le previsioni della partita l'endpoint utilizzato è "https://apifootball.com/api/?action=get_predictions". La risposta fornita dal servizio è:

```
[
  {
    "match_id": "3690090",
    "country_id": "253",
    "country_name": "Australia",
    "league_id": "684",
    "league_name": "A-League",
    "match_date": "2019-01-08",
    "match_status": "FT",
    "match_time": "08:50",
    "match_hometeam_name": "Western Sydney Wanderers FC",
    "match_awayteam_name": "Wellington Phoenix FC",
    "match_hometeam_score": "2",
    "match_awayteam_score": "3",
    "match_hometeam_halftime_score": "1",
    "match_awayteam_halftime_score": "1",
    "match_hometeam_extra_score": "",
    "match_awayteam_extra_score": "",
    "match_hometeam_penalty_score": "",
    "match_awayteam_penalty_score": "",
    "match_hometeam_system": "4-3-3",
    "match_awayteam_system": "3-4-3",
    "match_live": "0",
    "prob_HW": "20.00",
    "prob_D": "25.00",
    "prob_AW": "55.00"
  }
]
```

Per il parsing quindi è stata usata la seguente struttura dati:

```
struct Prediction: Decodable{
  let matchId: String?
  let probHw: String?
  let probD: String?
  let probAw: String?
}
```

Per ottenere le quote delle partite l'endpoint utilizzato è "https://apifootball.com/api/?action=get_odds". La risposta fornita dal servizio è:

```
[
  {
    "match_id": "148356",
    "odd_bookmakers": "10Bet",
    "odd_date": "2017-02-07 07:41:36",
    "odd_1": "1.77",
    "odd_x": "3.74",
    "odd_2": "5.30"
  }
]
```

Per il parsing quindi è stata usata la seguente struttura dati:

```
struct Odds: Decodable{
  let matchId: String?
  let odd1: String?
  let oddX: String?
  let odd2: String?
}
```

7 Software System Attribute

7.1 Reliability

As the majority of function requires an internet connection, the software is reliable as long as there are no connectivity problems.

7.2 Availability

The availability parameter also relies on the internet connection signal and on the responses provided by "https://newsapi.org" and "https://apifootball.com". No problems of availability were found both during the development phase and beta testing with users.

7.3 Security

The security of the iSport application was the main concern during the development. As all the information provided are retrieved from the web, there are different checks to perform in order to keep the user information safe. Furthermore, for the communication with external services, was chosen HTTPS protocol to guarantee greater security

7.4 Maintainability

The entire application is very maintainable as the code is entirely documented, in particular in several critical function. Therefore any developer who wants to improve it or make changes is able to do it without relevant difficulty.

7.5 Usability

The usability was another of our concern for the development. In order to improve it, a beta version of the application was given to 5 users and tested for 1 weeks. The result of this test highlighted usability issues. In particular, in the first version the buttons for changing information on the match detail page were not clearly visible. Another report concerned the lack of visual feedback after adding a quota to their ticket. Both of these problems were then solved in a later version of the application.

8 Test Cases

9 Cost Estimation