CSC 8930 Master's Project
Summer Semester, 2019

Mobile Application Development for Sports Applications

Khushboo Neema

COLLEGE OF
ARTS & SCIENCES

Agenda

- Introduction
- Technology Used
- Technical Contribution
- Dataset Design and Operation
- Augmented Reality
- Deeplinks & Notifications
- Google Mobile Ads
- Mobile Analytics
- Unit Testing
- Chromecast
- Other tools and Methodologies
- Demo
- Current & Future Work

Introduction

- The importance of mobile phones in our everyday life is undeniably unending, this is because of the huge transformation in mobile phone's operating system and storage which makes mobile application development is one of the most innovative and actively growing sectors.

- I am working on two different mobile applications which focuses on a sports domain specially basketball tournaments.

- They provide all the information related to players, team, games, coach, league subscription, playoff scores, stats, highlights, breaking news, game analytics and all the history of tournaments.

- This applications support apple TV, Augmented reality and Chromecast to watch live streaming on bigger size screen devices.

- Applications uses mobile analytics to track the activities of users, games and application behavior to enhance the business of the national tournament association.

Technology and Tools

Technologies

- Technology: iOS (iPhone/iPad/tvOS).
- Language: Swift 5.0.
- Web Services API: REST/SOAP
- Database: Coredata
- Virtual Reality: Augmented Reality
- Mobile Data Analytics
- Data visualization

Tools:

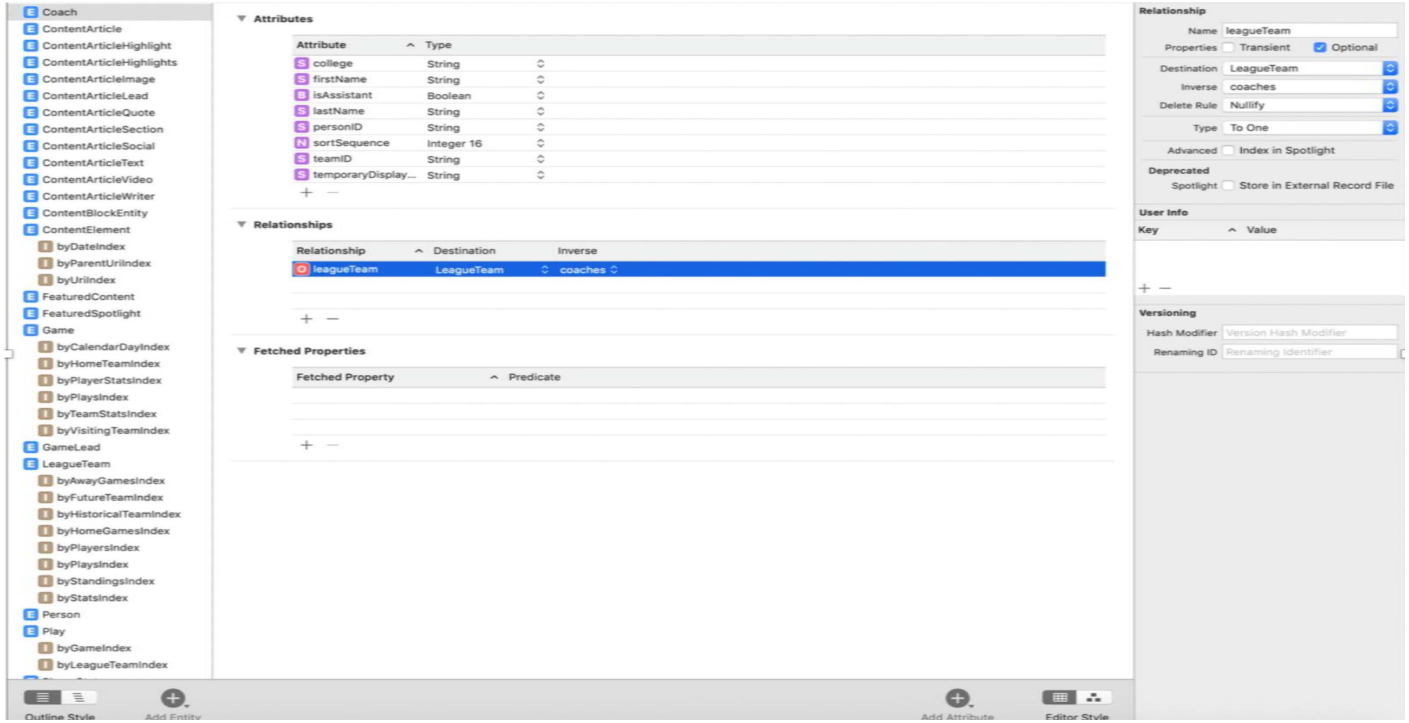
- Jenkins
- XCode
- Instruments
- JIRA
- SourceTree
- Slack

Database Design and its Operation

Coredata

- Due to limitation of less storage in mobile phones we cannot use any heavy database management system, So for mobile app we use Coredata which is small, fast, self-contained, highly reliable and fully featured

- Coredata is the object graph and persistence framework which create a database within an iOS application and allow relational-entity attribute model



- Coredata is the object-based model manages object graph rather than traditional table database methods. So, we create data model class for every tables in the core data and used its object to insert/update/delete and fetch queries in the Coredata

```
static func fetch(calendarDate: String, homeTeamTricode: String, awayTeamTricode: String,
                  gameNumber: Int, context: NSManagedObjectContext) -> Game? {
    let fetchRequest = NSFetchRequest<NSFetchRequestResult>(entityName: String(describing: Game.self))
    let predicate = NSPredicate(format: "calendarDate == %@ && homeTeam.tricode == %@ && visitingTeam.tricode == %@",
                                calendarDate, homeTeamTricode, awayTeamTricode)
    fetchRequest.predicate = predicate
    fetchRequest.fetchLimit = gameNumber

    do {
        let games = try context.fetch(fetchRequest) as? [Game]
        if let game = games?.last {
            return game
        }
    } catch let error as NSError {
        printError("Error in Game Creation/Fetch \(error.localizedDescription)")
    }

    return nil
}
```

- Designing Coredata is quite difficult than designing other relational database system because we can’t visualize the core data. To visualize tables and their entries after operations we use SQLite internally if possible.

Augmented Reality

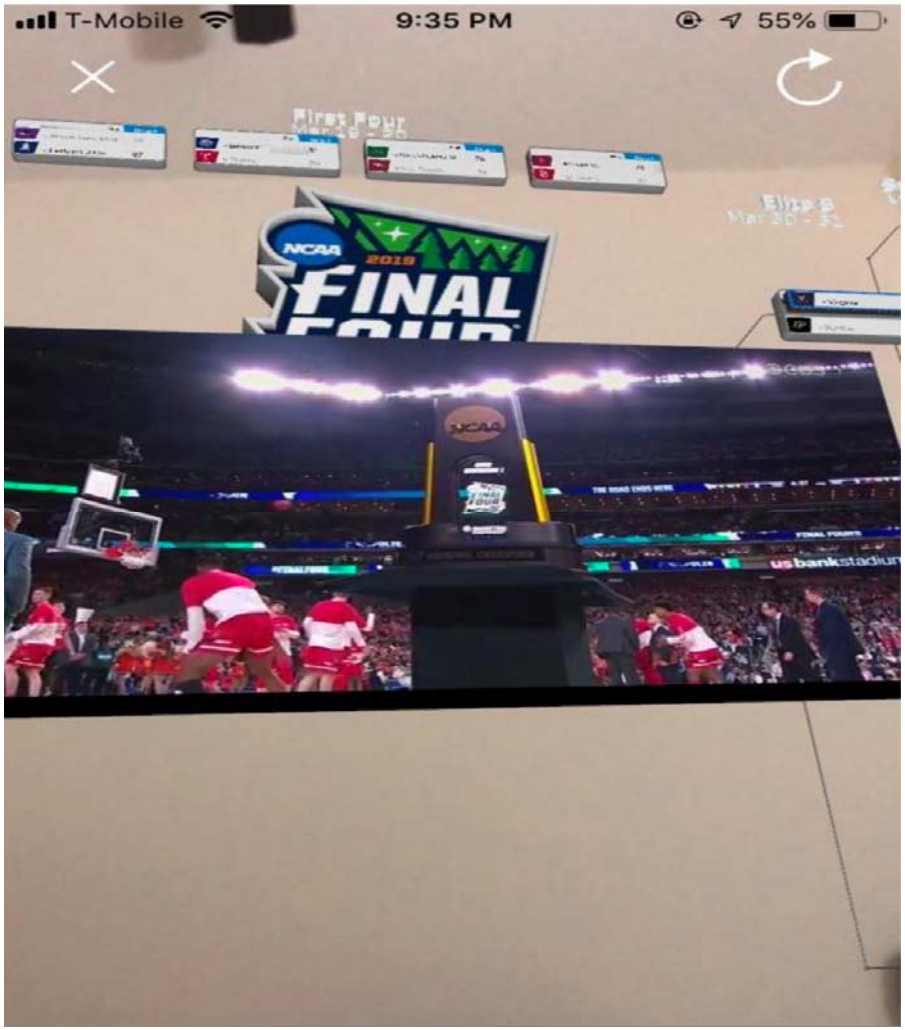
- The Augmented reality which is used to create simulated environment unlike traditional UI Interface

- Instead of viewing a simple UI on screen, users can immersed and able to interact with 3D worlds

- We have used the AR to show the brackets of the basketball game in the real-world environment wherever user is present and make it interactive with user. By a single tap User can watch any of the bracket's team game to the live or recap game. This make user feel like that they are watching game on a big screen television.

- To implement this we have used ARKit Sdk which is supported on all Apple devices that run iOS 11 and that have A11 (the newest), A10, or A9 chips

Augmented Reality



Implementation

To make AR applications we need to import the ARKit in our project and add ARKit's scenekit view to the application instead of simple view for user interface.

Camera is the important features that we use in AR enabled application, so we need to set camera permission to enabled for the application

We need to create every object's 3D image, for the brackets in the 3D view we have first created the real time bracket view programmatically in the application and convert this view first into image then convert it into 3D image with the extension. dae (digital asset exchange) format.

Once we'll have the 3D image we can programmatically cast this into AR virtual environment using available camera features in the phone

There are quite a few challenges in the implementation of Augmented Reality despite the advances in research and development area. The challenges are due to problems related to context-awareness, usability, navigation, visualization and interaction design.

Google Mobile Ads

Nowadays every business looking for ways to make extra money from the app. This is the most straightforward way is to put app in the App Store and sell it for $0.99 or more.

We have used Google mobile ads to display multiple ads within our application

Google mobile ads SDK provides API to add ads in the application in various formats (banner, videos or native ads) which best suits the user interface of the application

To create native ads style, we have written code in HTML, CSS, and JavaScript to define ads that are responsive and produce a quality display across all screens

To add ads in the applications we need to first register the application at google mobile ads website which generate the unique ID for our app then we need to import google mobile sdk and needs to set all the necessary setting in the project to support ads carefully

```swift
override open func execute() {
    var fileURL: URL

    if FileManager.default.fileExists(atPath: cacheFile.path) {
        fileURL = cacheFile
    } else {
        printInfo("No cached DFP config file. Falling back on bundled version.")

        // if a cache file is not available, we use the bundled version
        // swiftlint:disable:next force_unwrapping
        fileURL = Bundle(for: type(of: self)).url(forResource: "dfp", withExtension: "json")!
    }

    guard let stream = InputStream(url: fileURL) else {
        finish()
        return
    }

    stream.open()

    defer {
        stream.close()
    }

    do {
        guard let json = try JSONSerialization.jsonObject(with: stream, options: []) as? JSONObject else {
            throw NSError(domain: "nba.operations.dfp", code: 0, userInfo: [NSLocalizedDescriptionKey: "Invalid json at dfp endpoint"])
        }

        guard let dfpAdManager = try? Container.resolve(DFPAdManager.self) else { finish(); return }
        guard let
            adSlots = JSONParser.parseArray(json["adSlots"]) as? [JSONObject],
            let parameters = JSONParser.parseObject(json["globalAdSlotParameters"])
            else { finish(); return }

        dfpAdManager.adDictionary = parseDFPItemsFromDictionaryArray(adSlots)
        dfpAdManager.globalAdSlotParameters = parameters

        DispatchQueue.main.async(execute: {
            NotificationCenter.default.post(name: .dfpAdConfigFinishedParsing, object: nil)
        })

        finish()

    } catch let error as NSError {
        finishWithError(error)
    }
}
```

Mobile Analytics

- Mobile app analytics is the measurement and analysis of data generated when users interact with mobile applications
- It is the practice of collecting user behavior data, determining intent from those metrics and taking action to drive retention, engagement, and conversion
- Mobile Analytics helps the business to tell what users engage with, who those users are, what brings them to the site or app, and why they leave
- We collect all the user activity, what user doing inside the app, all his/her fav, dislikes and optimize it, apply analysis at local and pass the collected data to different servers depends on the requirements.
- Then business people use this data to find hidden trends and to take more informed decision.
- Our Client has very big multiple data analytics team, which uses most of the data to analysis generated by mobile traffic

Deeplinks and Notifications

- Deeplinks and Notifications are the main features of the mobile applications.
- Deeplinks can be access from the outside of the application, for eg. You are reading a news on the news app on the mobile, and you saw an link or advertisement regarding the app, if you press a single tab on it. It will directly take you inside the application at the same page you're looking for.
- Some common deeplinks are for games, players, teams, news, videos and etc
  link: gametime://players/202332, gametime://teams etc.
- Notifications are the very familiar term which keeps up up to date regarding our feeds and remainders.

Unit Test Cases

- Unit Testing is a level of software testing where individual units or components of a software are tested.

- It works in Test Driven Development environment. It is different from other form of automated and manual testing. It is also an automated testing, but it performs before the development of the modules begins.

- In our applications, initially we were not working in Test driven development environment which makes our efforts double in maintaining the application. So now we have adopt the TDD and we have In thousands of classes without unit testcases and for them we need to write the unit test cases

Unit Test Cases

```swift
class NotificationPreferencesTest: XCTestCase {

    private var notificationPref: NotificationPreferences!

    override func setUp() {
        // Put setup code here. This method is called before the invocation of each test method in the class.
        notificationPref = NotificationPreferences()
    }

    func testRegisterAndUnregisterTeamPreferences() {
        let teamId = makeRandomID()
        let teamNotification = TeamNotifications.allCategories(teamId)[0] as TeamNotifications

        notificationPref.registerTeamPreference(teamNotification)
        var teamPref = notificationPref.getRegisteredTeams()
        var isRegister = teamPref.contains(teamId)
        XCTAssertTrue(isRegister)

        notificationPref.unregisterTeamPreference(teamNotification)
        teamPref = notificationPref.getRegisteredTeams()
        isRegister = teamPref.contains(teamId)
        XCTAssertFalse(isRegister)

    }

    func testRegisterAndUnregisterGamePreference() {
        let gameID = makeRandomID()
        let gameNotification = GameNotifications.allCategories(gameID)[0] as GameNotifications

        notificationPref.registerGamePreference(gameNotification)
        var gamePref = notificationPref.getRegisteredGames()
        var isRegister = gamePref.contains(gameID)
        XCTAssertTrue(isRegister)

        notificationPref.unregisterGamePreference(gameNotification)
        gamePref = notificationPref.getRegisteredGames()
        isRegister = gamePref.contains(gameID)
        XCTAssertFalse(isRegister)
```

Chromecast

It is a streaming media adapter device generated by Google that allows users to play online content such as videos and music on a digital television.

We have provided a support of Chromecast devices to watch live or recap games on digital television.

To implement this feature, we have used integrate Google cast API into our project which cost $5 fee for Google cast developer registration

Google cast API has their own UI which we can reuse in our application, it also provides ability to customize application UI according to app's theme. We have used our own custom UI to display Chromecast connectivity.

Other Tools and Methodologies

Agile/Scrum Methodology :

- It helps the continuous integration of development and testing in the software development life cycle process.

- Agile breaks the product into smaller builds of multiple tickets(features) which mostly has to complete in 2 weeks of time span.

- Each ticket has a story point to complete the ticket. In this methodology there will be scrumMaster (Mostly will be a project manager of team lead) which helps scrum practitioner (developers) to achieve their highest level of performance within story points.

Tools :

- **SourceTree** - It is a powerful, simple and beautiful Git graphics user interface which simplifies how user interact with the git repository without distracting attention to coding

Tools

- **Jenkins -** It is the most famous continuous integration tool which is used to build and test software project continuously which makes it easier for developer to integrate changes in the projects.

- **JIRA -** It is a project management tool which is used to track bugs in the software project and mobile apps. It has a dashboard which consist of multiple functions and features to tract issues easily

- **Slack -** It is a tool which used to build and easy collaboration between you and multiple teams which situated at multiple locations.

Demo

Thank You

For Listening