**I. INTRODUCTION**

The **manager module - ARAds\_Manager\_Module** supports in the design augmented reality information to perform in presenter module. **Business managers** ***can*** create advertising campaign with augmented reality information, manage campaigns, and view statistics about users that have joined the campaign. **Administrators** ***can*** manage the whole system including customers, business managers as well as their campaigns.

**II. TECHNOLOGIES (Libraries and Frameworks)**

1. ASP.NET MVC5

2. Entity Framework

3. WCF services

4. Ninject

5. Bootstrap

6. Google Map API

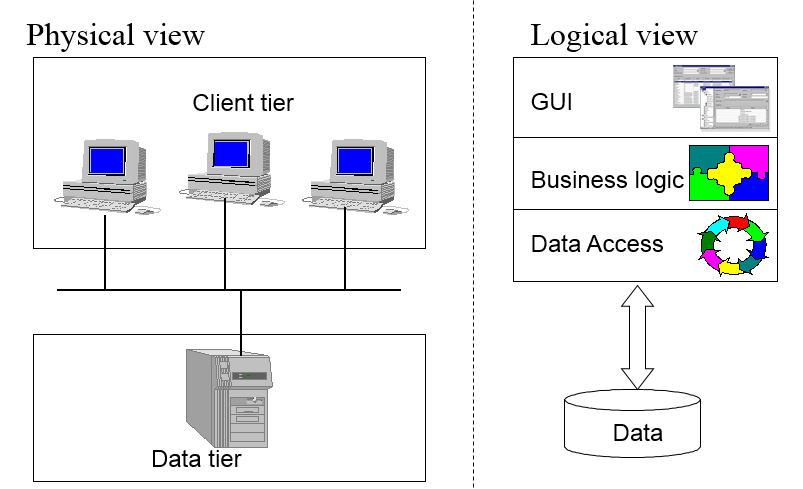
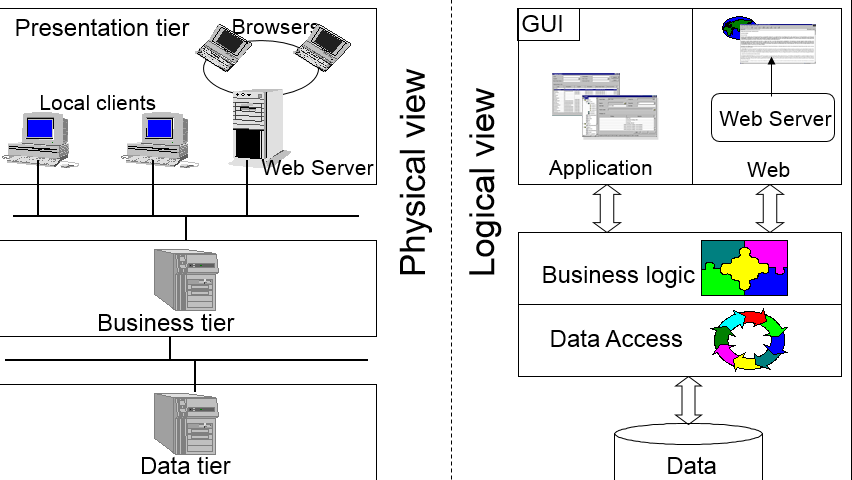
7. Facebook API

8. Android SDK

9. Vuforia SDK

**III. SOLUTION ARCHITECTURE**

Manager module is designed in **3 tiers architectures** containing **presentation tier**, **business tier** and **data tier** with associate layers that will be described later. The architecture increases performance, flexibility, maintainability, reusability, and scalability over 2 tiers.



**3 tier architecture 2 tier architecture**

**1. Data tier**

Data tier contain only **1 .mdf file** DT-DB1 – extension of MS SQL Server database file. The detail about structures and schema of the database will be describe in IV. Database design.

**2. Business tier**

The tier contains **2 layers**: Dao - ARAds.MM.Business.Dao (Data Access Layer) and Service - ARAds.MM.Business.Service.

**A. Dao** - ARAds.MM.Business.Dao

* + Project output type: C# class library
  + Libraries: Entity framework, JSON
  + The Dao layer plays **2 primary roles**:

\_ **Access** directly to database **DT-DB1** and use models classes in folder ***Models*** in Common - ARAds.MM.Common that has mapped with database by edmx file in Common to return data.

\_ Use models classes in Common that has mapped with database by edmx file in Common to **perform** **DML operations (insert, update, delete)** to database **DT-DB1**.

**B. Service** - ARAds.MM.Business.Service

* + Project output type: WCF application
  + Libraries: Ninject, JSON
  + The Service layer provides ***interfaces*** for clients (Web application and Presenter application) that are placed in folder ***Services*** in Common - ARAds.MM.Common to interact with database **DT-DB1**.
  + The implementation of services is store in Service - ARAds.MM.Business.Service.
  + **Exception** is caught and thrown in **Services** - ARAds.MM.Business.Service. The **concurrent update problem** is also handled at this layer.
  + Validate and convert data to work with Dao - ARAds.MM.Business.Dao.

**3. Presentation tier**

The layer is an ASP.NET MVC web application. The features of this module are

Provide GUI to users to use their rights that will be described later in V. Users & Use cases.

Catch exception throw by Services - ARAds.MM.Business.Service and access to in Common - ARAds.MM.Common to display associate error message to users.

Validate data from user input before calling services.

Validate data received by services before displaying to users.

**4. Architecture**

**WEB APPLICATION**

**PRESENTATION TIER**

**COMMON**

**SERVICES**

**DAO**

**BUSINESS TIER**



**DATA TIER**

**IV. DATABASES DESIGN**

**V. USERS & USE CASES**

**VI. APPENDIX**