

**COMP 3059 – Capstone Project I****Software Requirements Analysis and Design Assignment**

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:

[http://www.uacg.bg/filebank/acadstaff/userfiles/publ\\_bg\\_397\\_SDP\\_activities\\_and\\_steps.pdf](http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf)

[www.cse.msu.edu/~chengb/RE-491/Papers/SRSEExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSEExample-webapp.doc)

Source for this template:

[www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc](http://www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc)

## 1.0 Introduction

IRes is an electronic receptionist designed to help visitors and delivery personnel while reducing inefficiencies associated with human receptionist, wait times, and any errors. We will be using Android studios to complete this app and MongoDB to store data. Benefits of IRes: It will be keeping electronic visitor history, implementing MFA Authentication for visitors, providing real time business notifications, completing automatic delivery check in with custom code, and digitizing front desk interactions. All of this will significantly improve user and visitor experience.

### 1.1 Purpose

The purpose of this document is to describe the high-level software requirements for IRes. It will state what the system will do without discussing technological aspects. This document is intended for stakeholders, including business owners, IT support, developers, designers, and those who need a clear understanding of the system's capabilities.

### 1.2 Scope

The scope of **IRes** will define what the system will and will not do. **IRes** is an electronic receptionist that supports interactions between visitors and deliveries with companies/ businesses. It will reduce the chance of human error and reduce visitor check-in time by up to 50%, and delivery wait time by 60%, leading to an overall improvement in front desk efficiency.

The scope of **IRes** includes:

- **Visitor Check in**  
It's going to get information from the visitors on who they are meeting with, the floor they need access to (if applicable), phone number (for MFA Authentication), and the reason for their visit.
- **Delivery check-in** This will be based on a custom entrance code provided by the business or company.
- **Notifications to Business/Companies:**

There's going to be text notifications sent to the business or company letting them know if a visitor checked in or a delivery person was successfully granted access after inputting their custom generated entrance code.

- **Access Control:**

Businesses or companies will grant access to visitors through digital approvals. It will be sending generated custom code to the visitor's phone to implement MFA Authentication.

The system **will not be responsible for:**

- **Human Receptionist Tasks:**

IRes will handle tasks differently from a traditional human receptionist. It will not be bringing visitors to their destination, answering phone calls, or anything else a human receptionist will do.

- **Security Management:**

While IRes works in entry control, it will not replace building security systems like surveillance or security guards.

- **Hardware:**

IRes will focus on software and won't be responsible for making custom check-in kiosks or tablets.

The app will be available through the google play store on any mobile android device

- **Employee Check-In & Scheduling:**

IRes won't manage employee scheduling, or other HR related tasks.

**The primary goals and benefits of the IRes system are to:**

- Improve visitor and delivery check-ins.
- Reduce wait times for visitors and delivery personnel.
- Provide a user-friendly and secure interface.

## Process workflow

### Visitor Check-in Workflow:

- Visitors will approach the front desk where a check-in tablet will be.
- They will be choosing either visitor check-in or delivery check-in.
- After choosing visitor check-in they will have to answer some questions like:  
Who are they seeing, the reason for their visit, phone number, and the floor they need access to (if applicable).
- After submitting the form, it will be sent to a business or company via text message letting them know if someone is waiting for them. Then they decide if they want to grant access to the visitor by giving them a generated custom code with instructions.
- Visitors will receive digital access details directly to the visitor's phone number provided by the business or company if they have been approved. Then the visitor will have to input the generated custom code that was given to them into the tablet for them to get entry. If the code was right, it will grant access right away.

### Delivery Check-in Workflow:

- Delivery personnel approach the front desk where a check-in tablet will be.
- They will be choosing either visitor check-in or delivery check-in.
- After choosing delivery check-in they will have to input a generated custom entrance code that was given by the business or company.
- If the code was right, it will grant access right away.

### Objectives

- **Reduce Human Error:**

It will reduce human errors in the check-in process by allowing visitors and delivery personnel to check-in themselves instead of a human receptionist not knowing or understanding the purpose of their visit, leading to human errors or conflicts. This will lead to more accurate data collection and fewer mistakes.

- **Reduce Visitor Check-In Time:**

Reduce visitor check-in time by up to 50% through the check-in process, providing quicker and smoother access for visitors, and it will be safer for building management as well.

- **Reduce Delivery Wait Time:**

Lower delivery wait time by up to 60% by allowing them check-in with a custom entrance code given by a business or company beforehand instead of waiting for a response.

- **Improve Front Desk Efficiency:**

It will improve front desk efficiency by removing manual tasks and it will allow management to free up staff for more critical tasks.

The scope helps to stay within the project and not go over the boundaries.

## **2.0 System Overview**

### **2.1 Project Perspective**

This is a new self-contained system. The Electronic Receptionist system is a novel solution designed to automate and streamline front-office operations, enhancing efficiency and visitor experience.

## 2.2 System Context

- Streamline the visitor registration process, reducing manual paperwork and wait times.
- Provide a seamless and professional experience for all visitors.
- Gather insights on visitor patterns and preferences for future business decisions.

## 2.3 General Constraints

- The system must adhere to strict security standards and regulations to protect sensitive information.
- The system must be scalable to accommodate future growth and increased visitor traffic.
- The user interface should be intuitive and easy to use for both visitors and administrators.
- The system must be highly reliable and available during all active hours.

## 2.4 Assumptions and Dependencies

### Assumptions:

- *Companies will have access to electronic devices (smartphones, tablets, laptops, etc.).*
- *Stable internet connections will be available.*
- *Businesses will experience constant visitation.*
- *Users will be familiar with basic mobile and web applications.*
- *The operating device will remain stable and accessible.*
- Necessary hardware, such as tablets, will be procured and installed on time.

### Dependencies

- *The application must work across devices depending on what the individual location has.*
- *The application will need an internet connection.*
- *The software must be able to be made in 15 weeks.*
- *The application must be able to be used intuitively.*
- *The application must communicate with users (Text, onscreen displays, etc.).*
- Collaboration with security and IT teams for system integration.

## 3.0 Functional Requirements

### 3.1.1 Receiving Users

- Introduction: The software will interact with users (customers, delivery personal, etc.)
- Inputs: The system will take information like what business the customer is going to, phone number, time of the visit and the purpose
- Processing: The information is stored and formatted to be retrieved output

- Outputs: The system will output a confirmation text to the Business and if approved to the user

### **3.1.2 Viewing Data**

- Introduction: The software will store and format Information
- Inputs: The system will receive a request to view the information
- Processing: The System will access the database and then format the information
- Outputs: The system will display the formatted data

### **3.1.3 Approving Visitors & Deliveries**

- Introduction: The software will Notifies a designated person through text
- Inputs: The business will Provided a approve or refuse
- Processing: The System will respond based on the input
- Outputs: If successful the visitor will get a confirmation

## **3.2 Use Cases**

### **3.2.1 Use Case #1 Delivery**

- User selects the business they are delivering to
- Users enters Delivery as their purpose
- Business grant access to Delivery
- App sends message with confirmation
- Delivery is completed

### **3.2.2 Use Case #2 Customer**

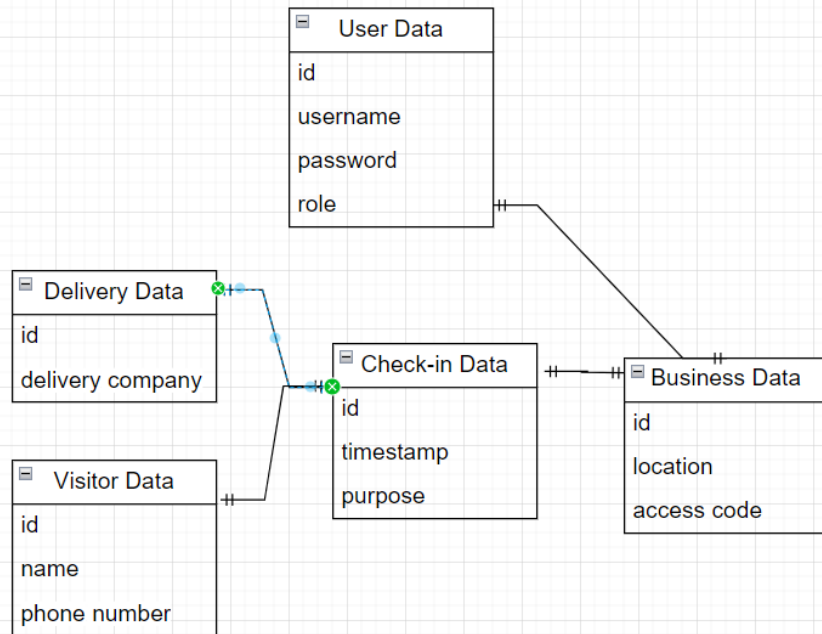
- User selects the business they are delivering to
- Users enters Visitor as their purpose
- User enters Name and Phone number
- Business grants access to the visitor
- App sends message with confirmation
- Visit is complete

### **3.2.3 Use Case #3 Admin Views data Collection**

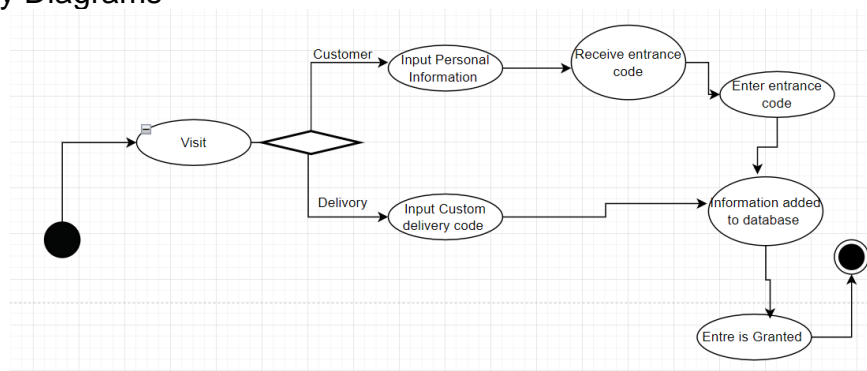
- Admin logs into app
- Admin selects Data collection
- App collects and formats Data

## **3.3 Data Modelling and Analysis**

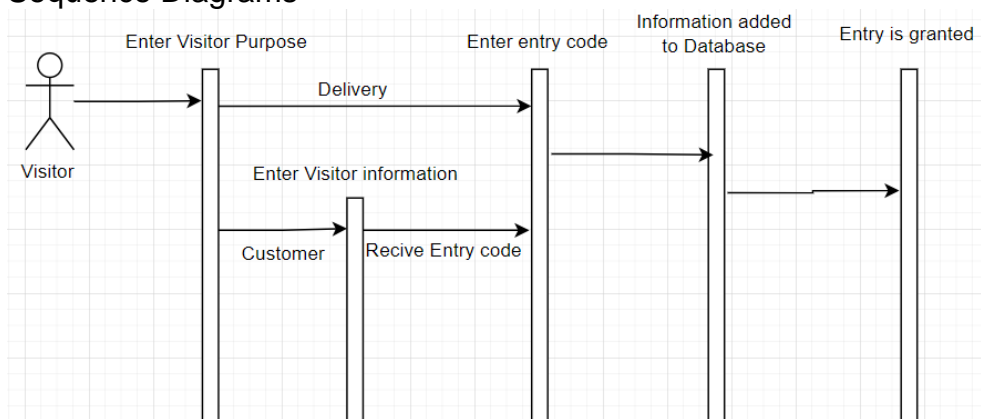
- Normalized Data Model Diagram



- Activity Diagrams

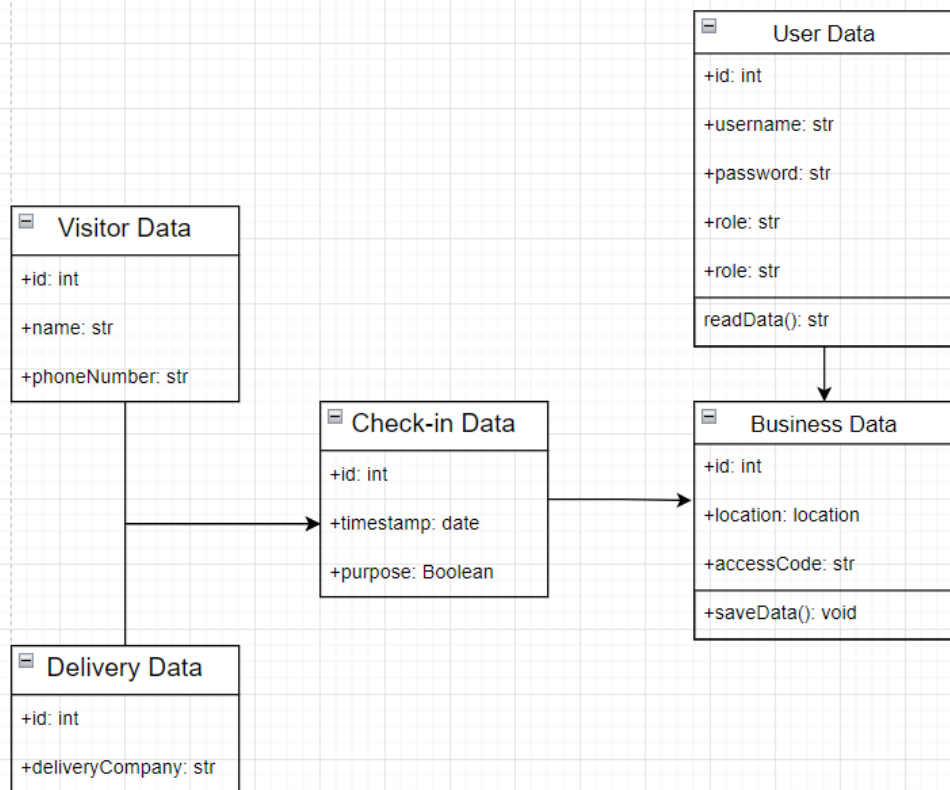


- Sequence Diagrams



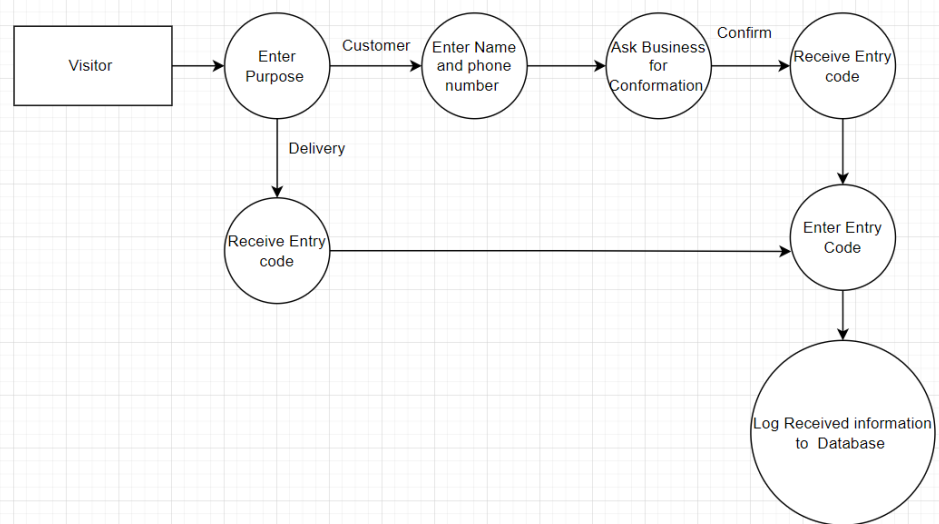
- UML Class Diagram





### 3.4 Process Modelling

- Data Flow Diagram



## 4.0 Non-Functional Requirements

### 4.1 - Performance

- Short loading screens for the main menu will take five seconds or less
- Loading screens for larger files such as documents will take 20 seconds or less

- Transaction throughput of 100 transactions per second to support concurrent users
- Reminders such as SMS notifications will arrive within 1 minute

#### **4.2 - Reliability**

- Provide clear feedback to users for invalid input in 99% of scenarios
- Create database backups every 15 minutes to prevent data loss
- Autosave document progress every minute while filled out by users in case of emergencies

#### **4.3 - Availability**

- 99% uptime during business hours, which can be set by the user

#### **4.4 - Security**

- Protection against 99% of SQL injections whenever users can enter code
- Prevent 99% of brute force attacks on the device with visitor permissions
- The app will create logs for actions involving user authority
- The app will encrypt data as it is sent between the app and database

#### **4.5 - Maintainability**

- Provide updates through the Google Play store, with little to no technical requirements required

#### **4.6 - Portability**

- Approximately 30% of the code is projected to be the Android interface frontend, which is not portable
- The remaining 70% of the code will be on the database backend, which will be portable

#### **4.7 - Usability**

- The user will be able to navigate the main menu to begin any task within 3 taps
- The app will provide responsive feedback to the user by using haptic feedback, on screen notifications, or text messages in 99% of actions
- Simple tasks such as user check-in should take no longer than 2 minutes
- More complex tasks such as filling out contact information should take no more than 5 minutes

#### **4.8 - Scalability**

- The database will be able to store thousands of entries which include visitors, business employees and administrators
- The database will expect no more than 100 people in the building concurrently

## 5.0 Logical Database Requirements

The app will implement NoSQL data (MongoDB) for the data records and support the logic of the application.

Data	Data fields	Data format	Data Retention	Data Integrity
Check-in	Id Time Stamp In, Purpose (Visit/Deliver)	Integer, Time stamp, Boolean	3 months	Accessed only by Building Admin
Visitor Data	Id, Name, Phone Number	Integer, String, String	3 months	Accessed only by Building Admin
Delivery Data	Id, Delivery Company	Integer, String	2-5 years	Accessed by building admin and businesses
Office/Business	Id, Location, Access Code	Integer, String, String	2-5 years	Access by building admin and only each business to be assigned the data of itself
User Data	Id, Username, Password, Role	Integer, String, String, String	2-5 years	Password are hashed, only accessed by building admin and business to certain extent

### Data storage capacity

- User count: assuming 100 visitors daily
- Data per Visitor: estimated at 2MB for each visitor

- Daily Data consumed: 200MB
- Annual Data: 73GB for the first year, the additional storage will be added for backup, redundancy, aiming for 100GB a year

**Backup and Redundancy**

- **Regular Backups:** Daily backups stored separately for disaster recovery.
- **Redundancy:** Ensure data is replicated across multiple locations to prevent data loss.

**Performance and Security**

- **Performance:** Use SSD storage for faster read/write operations.
- **Security:** Encrypt data at rest and in transit. Implement access controls and audit logs.

**6.0 Other Requirements****7.0 Approval**

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Backend	Margaret Terechtchenko	MT	2024-10-22
Backend	Tomer Edelman	TE	2024-10-25
Frontend	Berlean Gregori	BG	2024-10-25
Database	Ryan Tran	RT	2024-10-31
Research	Akeen Zhong	AZ	2024-10-31