|  |
| --- |
| RepairBud – Car Servicing Application |
| System Requirements And Design Document | |
| **Version 1.1** | |
| 10/11/2019 | |

# 

Table of Contents

[1.0 Introduction 3](#_Toc24547003)

[1.1 Purpose 3](#_Toc24547004)

[1.2 Scope 3](#_Toc24547005)

[2.0 System Overview 3](#_Toc24547006)

[2.1 Project Perspective 3](#_Toc24547007)

[2.2 System Context 3](#_Toc24547008)

[2.3 General Constraints 4](#_Toc24547009)

[2.4 Assumptions and Dependencies 4](#_Toc24547010)

[3.0 Functional Requirements 4](#_Toc24547011)

[3.1 System Environment 4](#_Toc24547012)

[3.2 Use Cases 5](#_Toc24547013)

[3.2.1 Access Daily Appointment Sheet 5](#_Toc24547014)

[3.2.2 Create Walk-in Appointment 5](#_Toc24547015)

[3.2.3 Inspect Customers Vehicle 6](#_Toc24547016)

[3.2.4 Send Inspection Sheet 7](#_Toc24547017)

[3.2.5 Open Sent Inspection Sheet 7](#_Toc24547018)

[3.2.6 Create New Repair Order 8](#_Toc24547019)

[3.2.7 Send Repair Order 8](#_Toc24547020)

[3.2.8 Set Status Of Repair Order 9](#_Toc24547021)

[3.2.9 Search For Customer Profile 9](#_Toc24547022)

[3.2.10 Create New Appointment 10](#_Toc24547023)

[4.0 Non-Functional Requirements 17](#_Toc24547024)

[5.0 Logical Database Requirements 17](#_Toc24547025)

# 1.0 Introduction

## Purpose

The purpose for this document to outline system requirements and data models for the RepairBud – Car Servicing Application. It will present readers with the full scope of the system to be implemented. Such outlining details will ensure the reader with a full understanding of the systems features, software requirements and database models.

This documentation will provide a detailed overview of our software we are implementing. It will present the products parameters and goals. It presents a visual concept on how our team sees the product and its functionalities. It will aid in the software delivery lifecycle processes by defining all data models needed to ensure a smooth transition to implementation phase.

## Scope

The purpose for building the RepairBud – Car Servicing application points to the lack of innovation in the Car Repair Industry of big brand car dealership. The software being developed by the T15 group will help solve the problem of paper based customer vehicle progressing currently in place at car servicing dealerships. The general purpose of the software is to ease customer information management and to create a convenient and easy-to-us application for employees in these dealerships. It will allow users to inspect customers vehicles digitally, update a customer’s profile, book car servicing appointments, and allow users to create the repair orders all in one system, with a comfortable user experience.

# System Overview

## Project Perspective

The Software to be implemented will be a pilot project for the Classic Honda Dealership. It will be use to cover a wide array of issues currently in place with the processing of customers information. The software to be developed will replace all paper-based business processes in the Classic Honda Dealership.

Trying to cut back on paper-based documentation and easing into the digital age. The T-15 development group felt this would a good oppurpority to launch there pilot software, as well as develop a templated business ventures to another dealerships or repairShop looking for a more secure way to process customer information.

## System Context

The RepairBud application with help servicing teams with processing customer information quicker and easier by digitalizing the current procedures in place. It will move fluently from one process to the next. From checking-in the customers, to preparing and storing repair order, to checking-out the customer and booking there next desired appointment. It will move in a fluent direction eliminating the chance of human error and the risks associated with the use of paper currently in place. The system will be ran and developed for a variety of platforms ranging from hand-held devices, and desktop application.

The RepairBud application is only for internal company employees where customers of that establishment will not interact with the system.

## General Constraints

The timeframe of implementation and deployment set of 8 months, starting from September 2019

Stakeholders are expecting 4 agile sprints for presentation and updates

Technologies are limited to the teams personal devices

Testing of software will be limited to availability of staff members of the Classic honda service department

## Assumptions and Dependencies

RepairBud car servicing application will be a software developed in the delay time span of 8 months.

End-users will be present for testing during implementation of the software through various phases.

The development Group will perform implementation and planning with the agile methodologies

Company will write the solutions in the Java language

Testing will be dependent on the availability of Classic Honda Staff

A Remote repository will be create to store source code and packages only available to the team member

## 3.0 Functional Requirements

### 3.1 System Environment

The RepairBud – Car Servicing Application has 4 actors interfacing without software. These Actors consist of the Service Concierge, Service Advisor, Service Coordinator, and the Service Manager. These actors will be sending input such as customer information, customer vehicle information, Repair Work Orders, Appointment Information etc. The system will output information such as customers profile and appointment list.

## 3.2 Use Cases

This section present readers with use case processes for individual actors and the use cases that they are associated with. It will go into full description of the use case and outline the flow and business process.

### 3.2.1 Access Daily Appointment Sheet

**ID:** 1

**Actors:** Service Concierge, Service Advisor, Service Coordinator, Service Advisor

**Description:** During Initial login of the system regardless of what permissions you have as users, users will be presented with the current work days appointment list

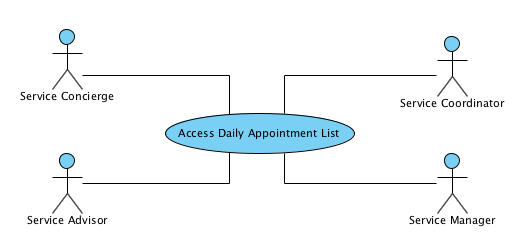
**Organizational Benefit:** Will help organize the current days work progresses and allow employees to gauge the intensity of the day ahead

**Triggers:** Users has selected the ‘Appointments’ Tab in the navigation bar or initial login to the system

**Basic Flow:**

1. Use Case starts on initial login or when user clicks ‘Appointment’ Tab
2. User Is Presented with the current days appointment list
3. Users can Then filter out appointment list in intervals (hourly)
4. Users Have the option to view a specific date when calendar icon is selected next to the date

**Diagram:**

****

### 3.2.2 Create Walk-in Appointment

**ID:** 2

**Actors:** Service Concierge

**Description:** If an customer does not have an appointment on the list, user will be able to create a Walk-In Appointment

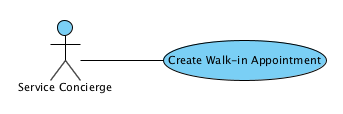
**Organizational Benefit:** Will help the organization service customers that are looking for car servicing during a walk-in situation

**Triggers:** Users has selected the ‘Create Walk-In Appointment’ At the top of the appointment list Screen

**Basic Flow:**

1. Use Case starts when user selects ‘Create Walk-In Appointment’ At the top of the appointment list Screen
2. User Is Presented with a form to create a New Customer Profile Or They can use the search car to find a customer profile if they already exist
3. Users Will then go through the car inspection process (3.2.3)
4. Once inspection process is done the user will click create
5. The walk-in appointment is when added to the appointment list with a ‘Walk-In’ label next to the appointment slot

**Diagram:**



### 3.2.3 Inspect Customers Vehicle

**ID:** 3

**Actors:** Service Concierge

**Description:** Once a customer appointment slot is clicked in the appointment screen user will be able to go through the vehicle inspection process

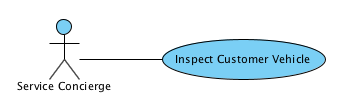
**Organizational Benefit:** Will help the organization keep vehicle inspection information digital and reduce the risk associated when process was done through paper documentation

**Triggers:** Users selects a appointment slot in the list of appointments on the appointment screen

**Basic Flow:**

1. Use Case starts when user selects a appointment slot in the list of appointments on the appointment screen
2. Users will be presented a series of tabs labelled ‘Customer Vehicle Information’, ‘Inspection Sheet’, ‘Services’ and ‘Customer Confirmation’. Process will start are Customer Vehicle Information
3. Users can Edit Vehicle Information
4. Users Can then Click the Inspection Sheet
5. Users can Take Pictures of the Customers vehicle when they click the camera icon
6. Users can Take a video of the customers vehicle when they click the video camera icon
7. Users can uses a touch template of a car to note down vehicle scratches
8. Users then Click ‘Services’ Tab
9. Users can scroll through a checklist of variable services that the repair shop offers
10. Users will Then Click the ‘Confirmation’ Tab
11. Users will be presented a declaimer provide by the company and a digital signature box
12. Users will then input a signature and move to function 3.2.4

**` Diagram:**

****

### 3.2.4 Send Inspection Sheet

**ID:** 4

**Actors:** Service Concierge

**Description:** Once an inspection sheet is complete users will then filter through a dropdown list of advisors to send the inspection sheet along with customers profile

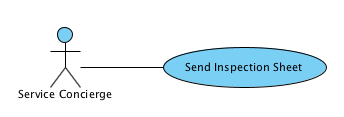
**Organizational Benefit:** Will help the organization keep the business process moving smoothly through one process to the next

**Triggers:** Users are on the ‘Confirmation’ Tab of the inspection sheet

**Basic Flow:**

1. Use Case starts when user are on the ‘Confirmation’ Tab of the inspection sheet appointments on the appointment screen
2. After users click finish on the confirmation page they will be presented with window prompted with a dropdown list of advisors
3. Users will click the advisor to send the inspection sheet and profile too
4. Users will hit send
5. Users will be present a confirmation message prompt

**Diagram:**

****

### 3.2.5 Open Sent Inspection Sheet

**ID:** 5

**Actors:** Service Advisor

**Description:** Inspection sheets from the Service Concierge is presented in a Service Advisors Inbox

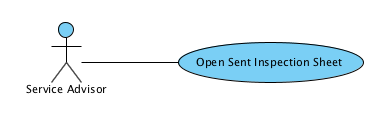
**Organizational Benefit:** Will help the organization keep the business process moving smoothly through one process to the next

**Triggers:** Users open an inspection sheet that was sent to them by Advisor

**Basic Flow:**

1. A notification will prompted to an advisors screen when a concierge sends then an inspection sheet
2. Users can click the mail icon on the top of the navigation bar
3. Users are presented with a mail system
4. Users can click the inspection sheet that was sent to them
5. Users will then be presented with the same series of tabs in the 3.2.3 list but information is non editable and the confirmation Tab is replaced with a ‘Repair Order’ tab
6. Users can navigate through the tabs and view the information

**Diagram:**

****

### 3.2.6 Create New Repair Order

**ID:** 6

**Actors:** Service Advisor

**Description:** Users will be able to create the Repair Order the technicians uses that allows them to complete the specific task that were for the service

**Organizational Benefit:** Will help the organization keep the business process moving smoothly through one process to the next

**Triggers:** Users Click the ‘Create New Work Order’ at the bottom of the inspection tab

**Basic Flow:**

1. Use case starts when users click the ‘Create New Work Order’ in the ‘Repair Order’ tab of the sent inspection sheet
2. Users are presented with a form where customer information and vehicle information are auto filled based on the customer information
3. Users can Add special notes for the technicians
4. Users can Add the specified services that the customers wants
5. User then click ‘Save’ at the bottom of the work order

### 3.2.7 Send Repair Order

**ID:** 7

**Actors:** Service Advisor

**Description:** Users will be allowed to send the details of a repair order in PDF form to a technicians for details of work to be done

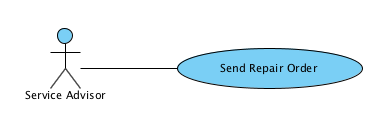
**Organizational Benefit:** Will help the organization keep the business process moving smoothly through one process to the next

**Triggers:** Users clicks finish at the bottom of the ‘Repair Order’ Tab

**Basic Flow:**

1. Use case starts when users click the ‘Finish’ bottom of the ‘Repair Order’ Tab
2. Users are presented a window prompt with a dropdown list of technicians
3. Users click the desired technicians to send the repair order to
4. User clicks ‘send’
5. User will be presented a confirmation Message

**Diagram:**

****

### 3.2.8 Set Status Of Repair Order

**ID:** 8

**Actors:** Service Advisor

**Description:** Users will be allowed to toggle a ‘status’ report next to opened repair order

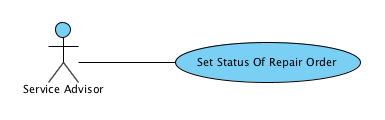
**Organizational Benefit:** Will help the organization keep repair orders organized and track status of a customer vehicle

**Triggers:** Users click the ‘status’ dropdown list next to a opened repair order in the repair order list

**Basic Flow:**

1. Use case starts when users click the ‘status’ dropdown list next to a opened repair order in the repair order list
2. Users will be presented a dropdown list contain status messages such as ‘Waiting For Technician’, ‘Work In Progress’, ‘Finished – Invoiced’ User clicks ‘send’
3. Users will Click one and depending what option is clicked will have a corresponding color

**Diagram:**

****

### 3.2.9 Search For Customer Profile

**ID:** 9

**Actors:** Service Concierge, Service Advisor, Service Coordinator, Service Advisor

**Description:** Users will be allowed to search for a specific customer Profile based on Vehicle Identification Number, Phone Number, Address, Email etc.

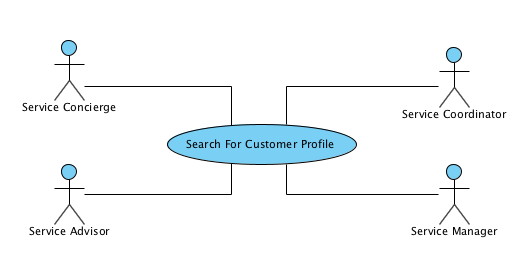
**Organizational Benefit:** Will help the organization quickly access a customer profile

**Triggers:** Users click the search bar

**Basic Flow:**

1. Use case starts when users clicks the search bar
2. Users will be Ables to input data to search up a customer profile
3. Based on the inputted data it will filter out results for the specified query and present the options found to the user
4. User will click on the profile deemed desirable and will be presented with the customer profile

**Diagram:**



### 3.2.10 Create New Appointment

**ID:** 10

**Actors:** Coordinator

**Description:** Users will be allowed to create a new customer booking for a car servicing appointment

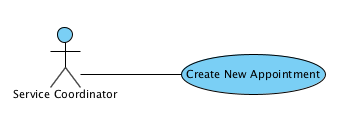
**Organizational Benefit:** Will help the organization quickly access a customer profile

**Triggers:** Users click ‘Add New Appointment’ Button at the top of the appointments tab

**Basic Flow:**

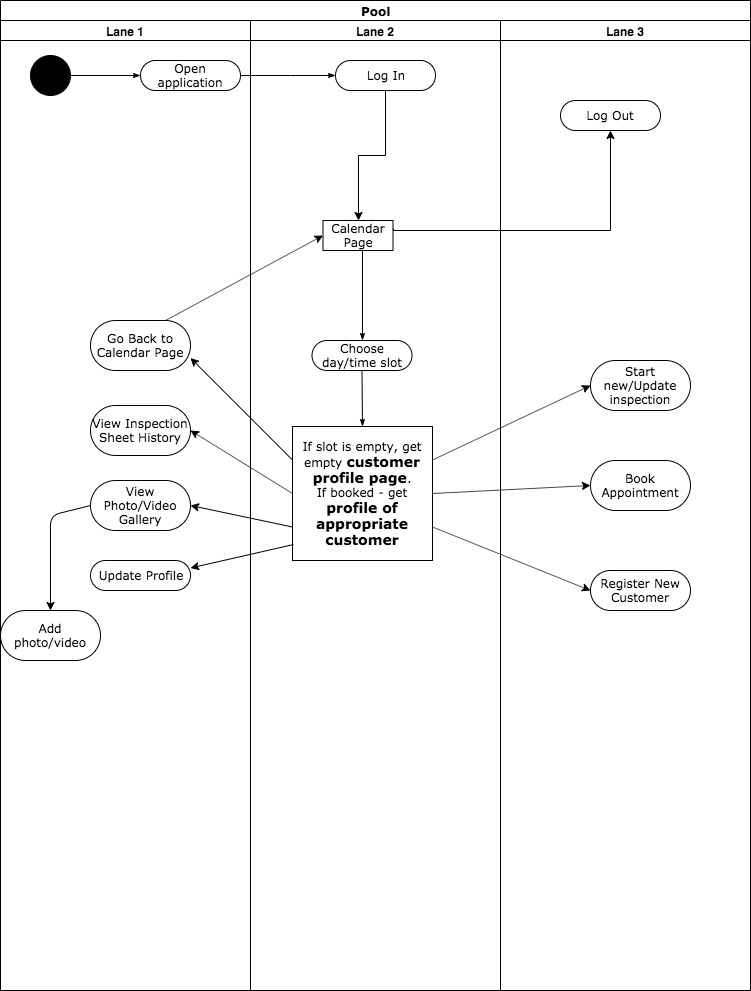
1. Use case starts when Users click ‘Add New Appointment’ Button at the top of the appointments tab
2. Users will then be presented a form
3. User will include the customers profile to add the booking for
4. User will then pick a date the booking is for with a calendar icon
5. User will then add specific notes or concerns the customer has
6. User will click ‘Add Appointment’
7. User will be presented a confirmation Screen

**Diagram:**

****

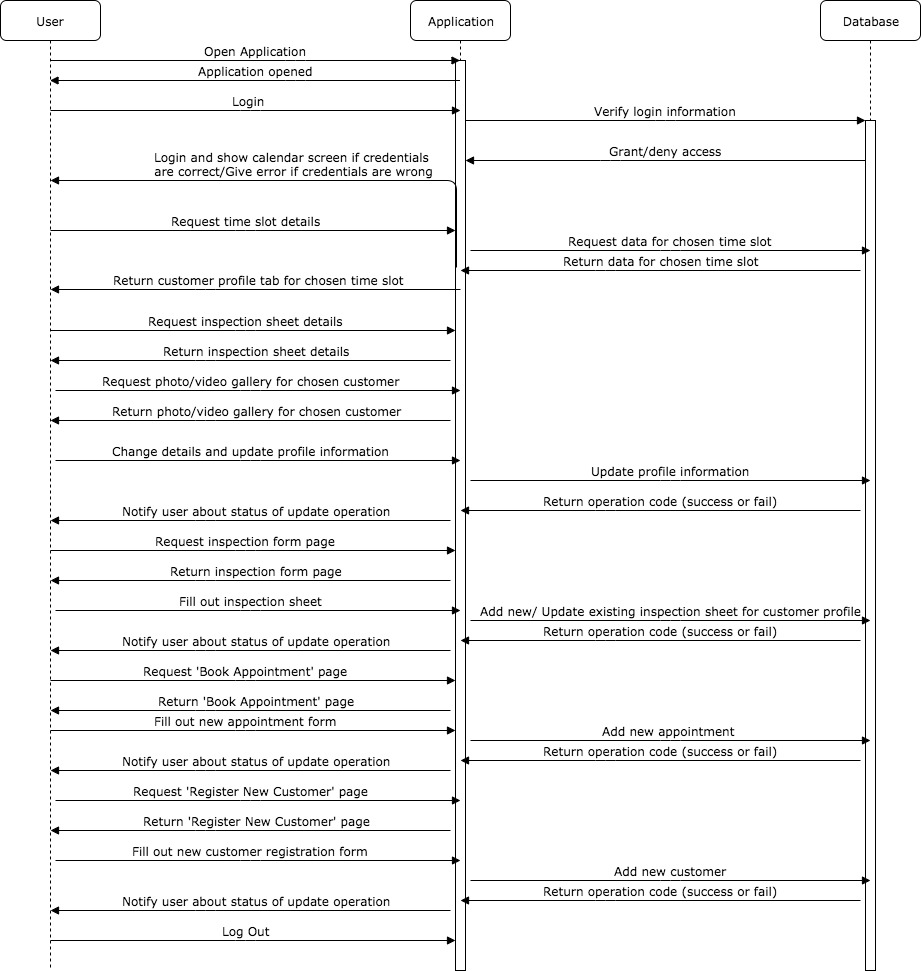
**3.3 Data Modelling and Analysis**

**3.3.1 Activity Diagram**

****

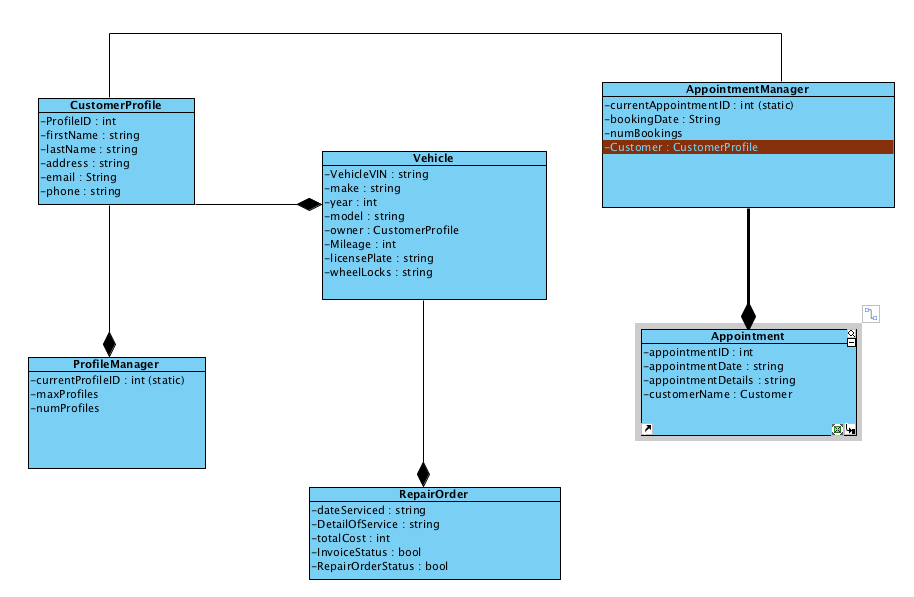
*Figure 3.3.1 the diagram displayed depicts the sequence of events raised in our software when a user must inspect and update a customer’s profile based on whether the customer had a scheduled appointment in the appointment list or not*

**3.3.2 Sequence Diagram**

****

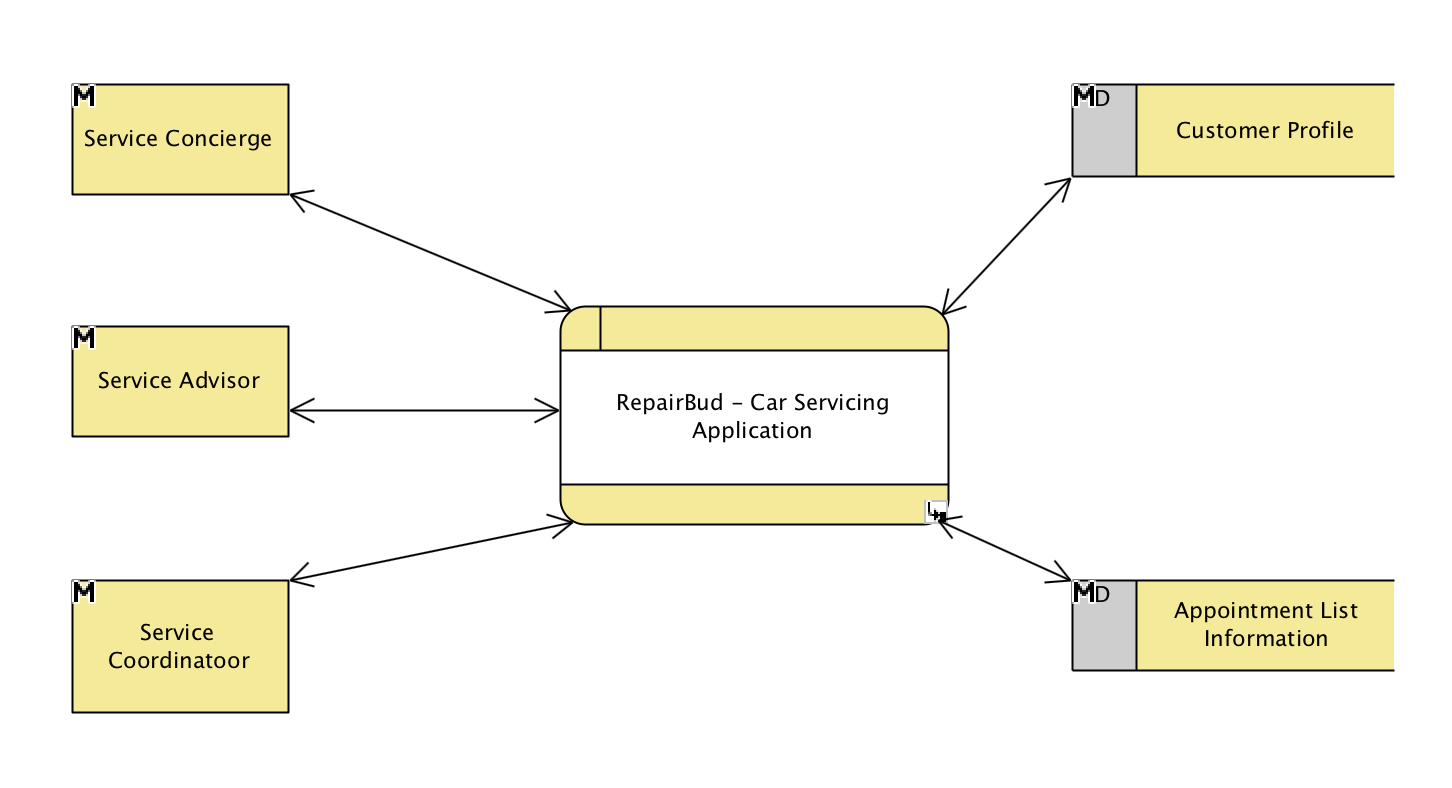
*Figure 3.3.2 represents the stages of user input with the software and the outputting software responses raised by our software*

**3.3.3 UML Class Diagram**

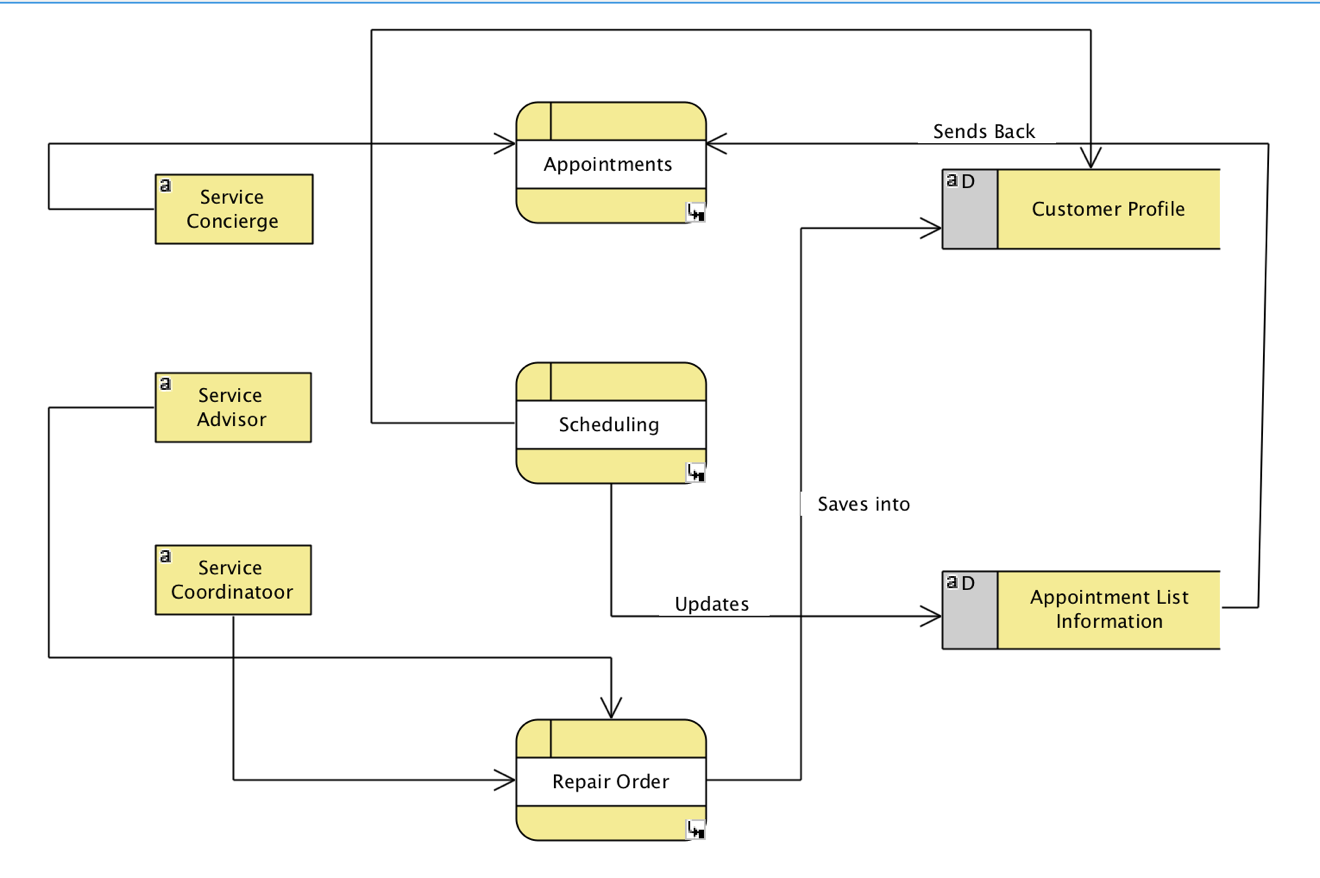


*Figure 3.3.3 this figure depicts the different classes our software team will be implementing into our code*

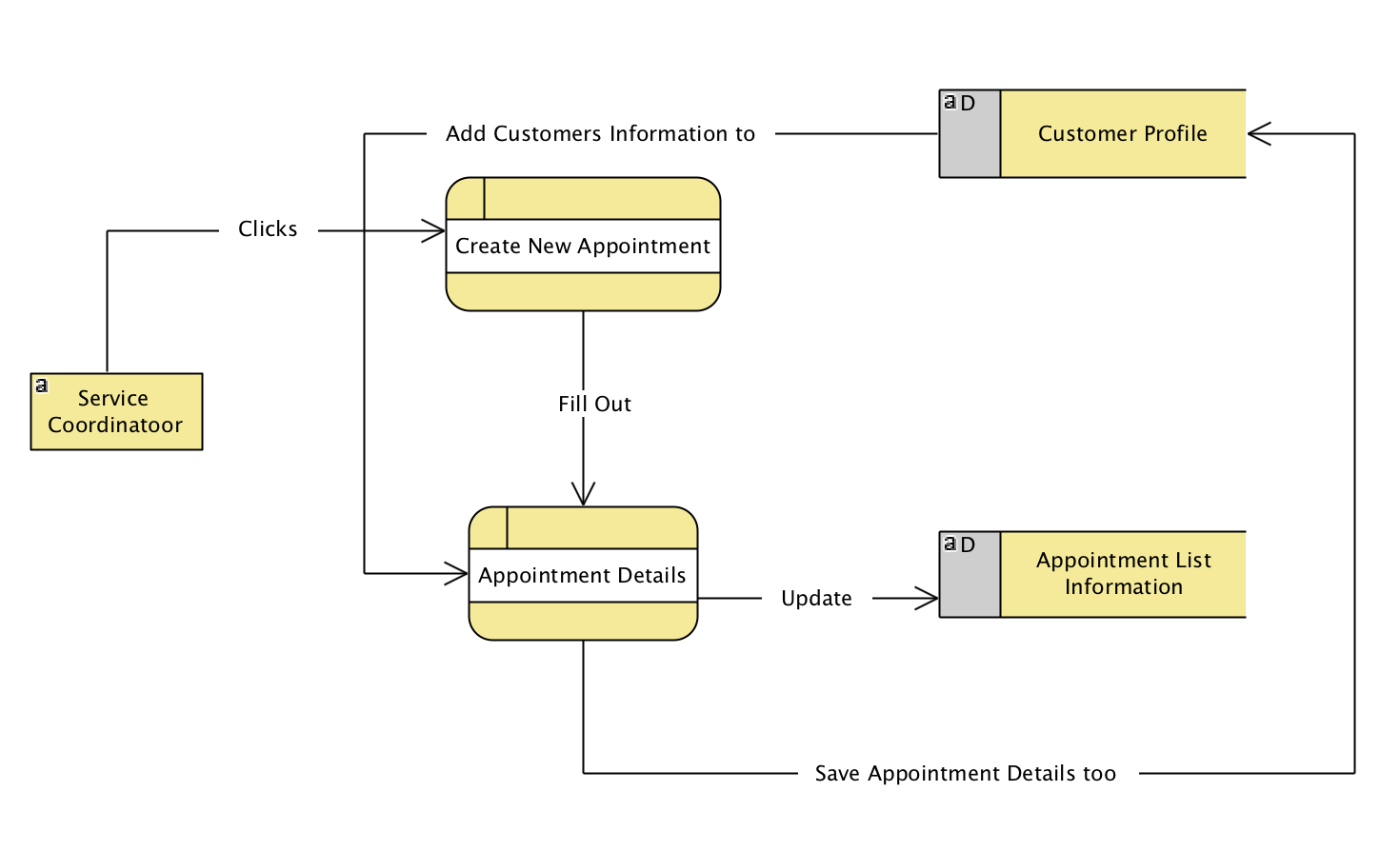
**3.4 Process Modelling**

**3.4.1 Context Data Flow Diagram **

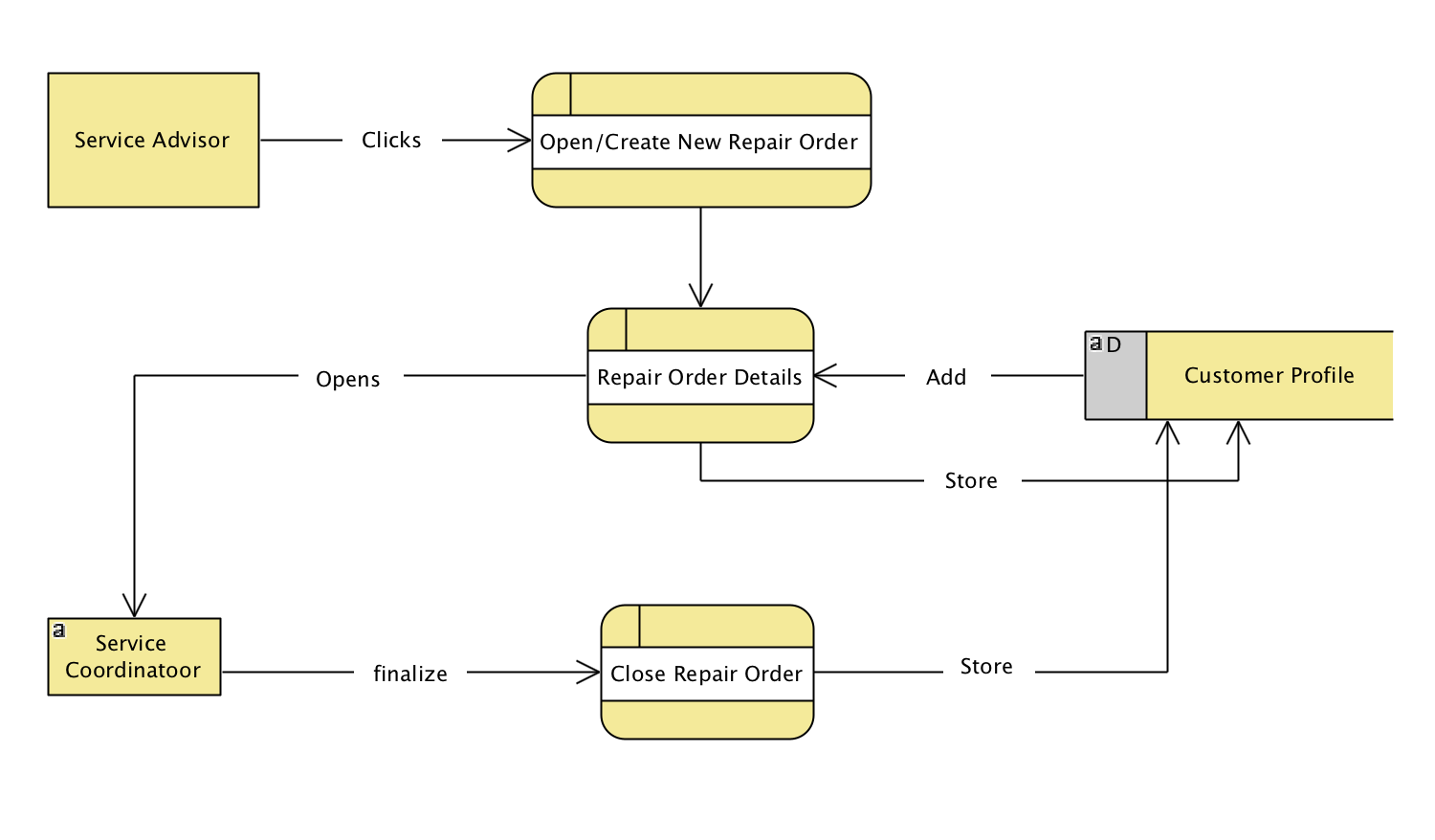
**3.4.2 Level 1 System Data Flow Diagram**

****

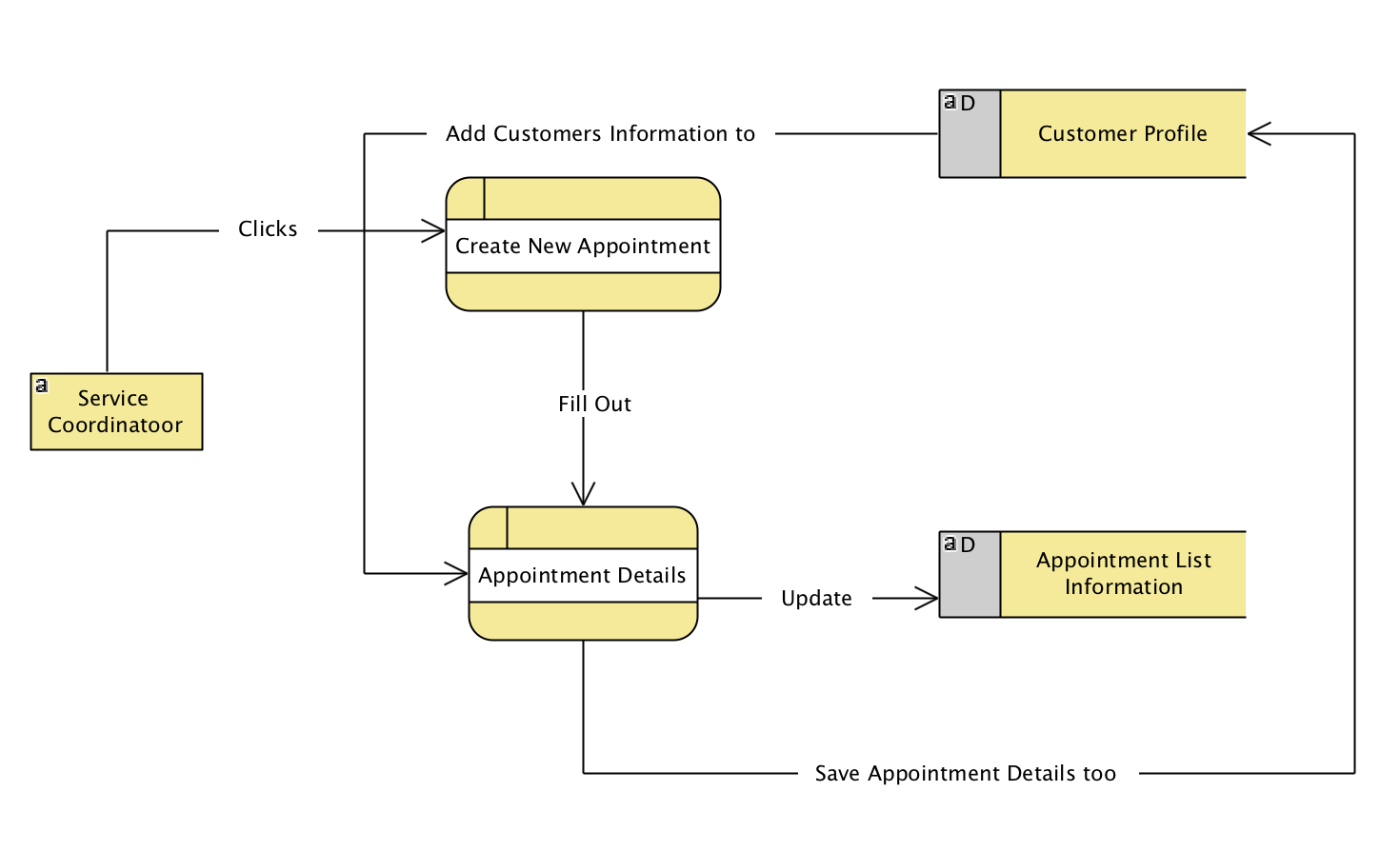
**3.4.2 Level 2 Data Flow Diagram – Check-in Customer**

****

**3.4.3 Level 2 Data Flow Diagram – Create Work Order**

****

**3.4.3 Level 2 Data Flow Diagram – Create New Appointment**

****

## 4.0 Non-Functional Requirements

**Availability/Reliability**: System is required to have 100% availability

**Appearance/Usability**: All pages and tabs have to be clear and easy to use/navigate

**Speed/Performance**: All pages and tabs but customer profile (as it requires to upload data from database first) have to load instantly

**Security**: all confidential customer information have to be secure according to industry best practices

## 5.0 Logical Database Requirements

**5.1 File Format**

The system must store all the Customer Profiles, Appointment Catalogue as well as. Data stored in the software will be in a text-based file. Users of our software, mainly repair shop employees, and all there login and personal information will be stored in one file. Attribute’s resided in the database will be delimited by a semicolon. Entries in the database will be alphabetically order arrange by specified unique ID depending on the table that is to be fetched. For each user interacting and active in the software a report card like file will be keeping track of session information. Attributes recorded in this report card will be as followed; session ID, date, duration, profiles edited, appointments booked. This will help gauge user activity and collect session information.

**5.2 Security**

The database server which all business data will be store will have security measures in place to ensure unauthorized read/write/delete access. Customer Profiles and confidential information stored will be encrypted and users registered to the software will be granted access to these files. The system will keep a log file of any access into precious files to ensure targeted usage. System will output error messages/alerts when data files validity is risked (invalid data, un-granted file access etc.)

**7.0 Approval**

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

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
| Developer | Artem Dryevov |  | 10/11/2019 |
| Developer | Patrick Parreno |  | 10/11/2019 |
| Developer | Asim Patel |  | 10/11/2019 |
| Developer | Elham Salaminam |  | 10/11/2019 |