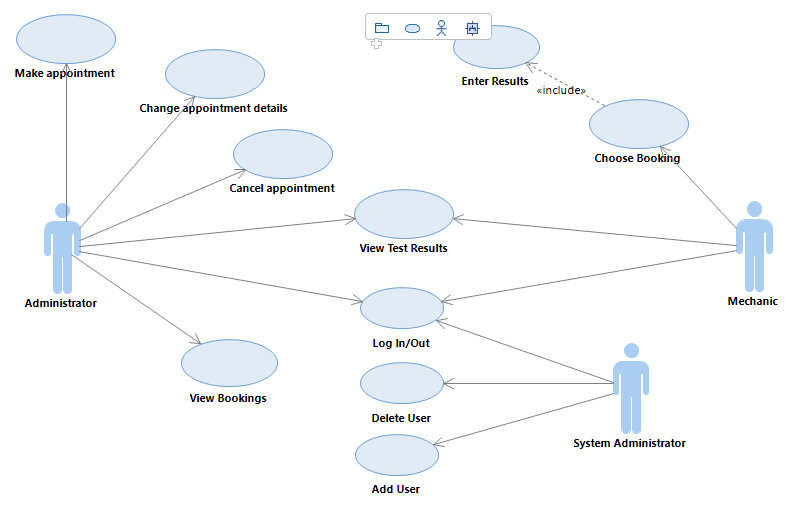
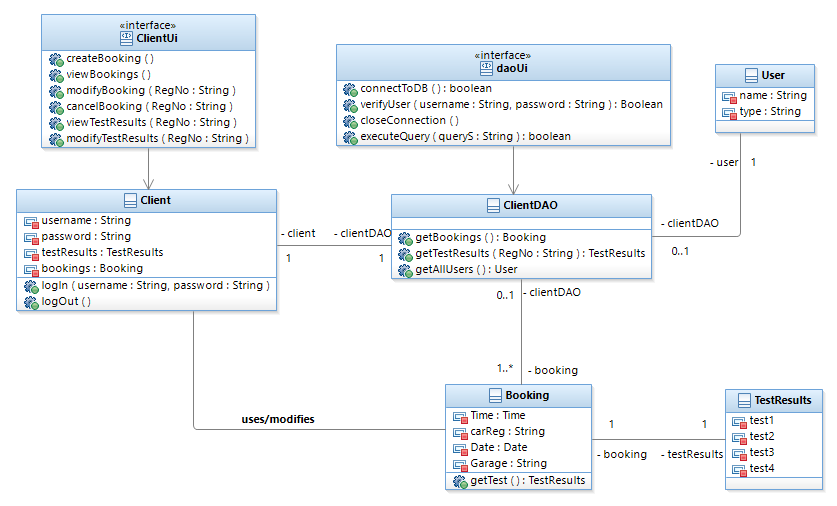
|  |
| --- |
| Project Title NCT Management System |
| Version Number: 1.0 **Change History**   |  |  |  | | --- | --- | --- | | **Date** | **Author** | **Comments** | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
| Summary Our system is one for garages to use to receive, organise and carry out NCT bookings and store the results. The system comprises a desktop application and an Android application that both share the same database. The bookings can be viewed on both the desktop application and the Android application by date.  The garage manager acts as the administrator for the system and only he/she has the ability to add or delete bookings. The mechanic only has the ability to view bookings and carry out NCT tests. When a car is selected from the list for the given day, the criteria for the test will be displayed with pass or fail criteria for each option. Upon completion of the test, the results will be sent to the database and the car list in the database will be updated. |
| Assumptions & Dependencies The desktop application is not operating system specific however the latest version of Java must be installed on the computer that will host the application. This computer must also have internet access in order to access the database to be used.  The mechanic must also have access to an Android device with a minimum of version 3.0 installed. The device must also have internet access. |
| Development Methods The desktop and Android applications will both be written in Java. Eclipse will be used to develop both applications. The database will be written in MySQL and will be hosted on a virtual machine using the Okeanos service. The virtual machine will have a full LAMP stack configured on it. All code will be shared in an organised manner using a private GitHub repository |
| High Level System Architecture The Okeanos virtual machine will be running as a server with a full LAMP stack running on it. As a result of this, the virtual machine and the database hosted on it can be accessed using the IP address of the machine. The machine is hosted on the cloud and as a result of this, will be running constantly in the background.  The Java application can be used by both administrators and mechanics. Upon login, the user who logs in will be displayed different information based on whether they are a mechanic or an administrator. This is because they have different access privileges within the system. The Android application works in the same way. |
| Detailed System Design   These are included at the end of the document due to formatting difficulties.  0. Use case diagram  1. Class diagram  2. Sequence diagram  3. Database design  4. GUI prototype  5. Information architecture (web sitemap) |
| Related Documents  |  |  |  | | --- | --- | --- | | Document Title | Author(s) | Description | | *e.g. test plan, design document, user manual, test results* |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |

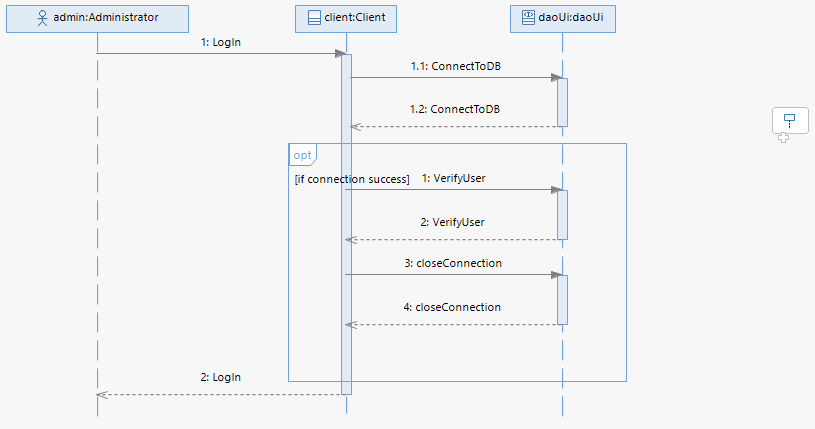
1. Use case diagram

2. Class diagram

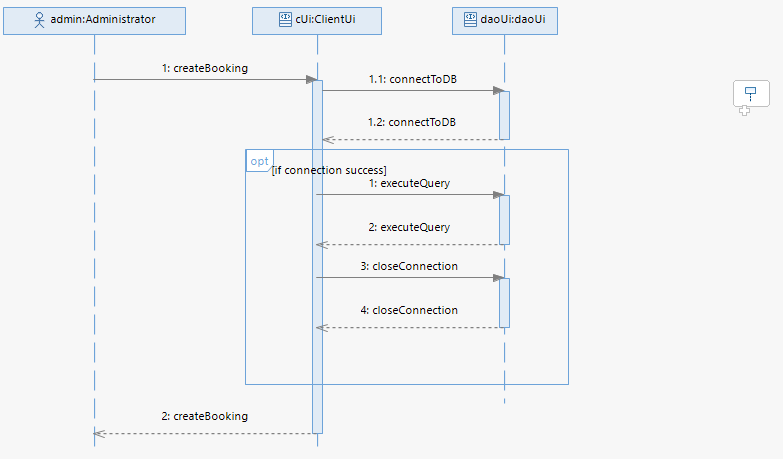


3. Sequence Diagrams

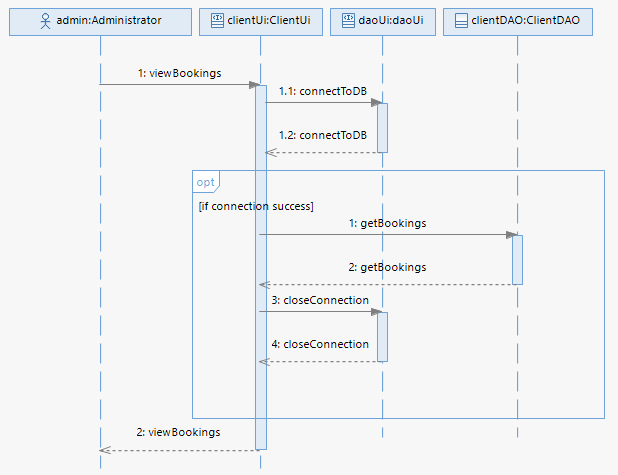
Login



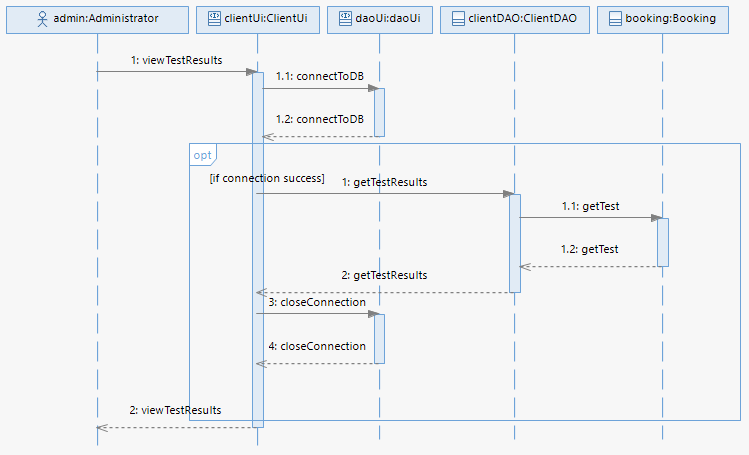
Make/Modify Booking



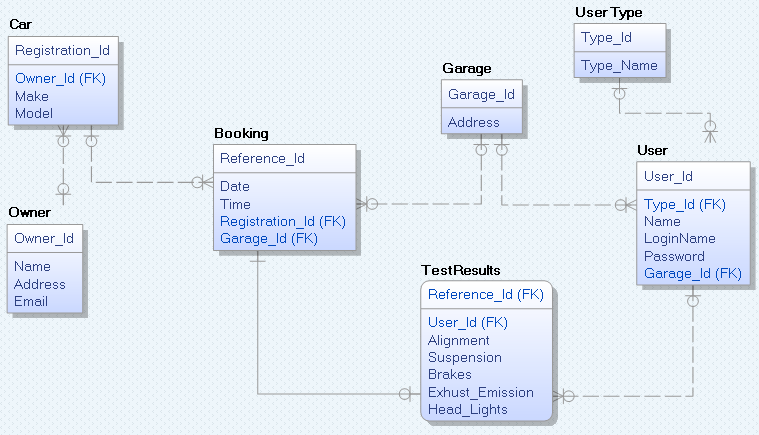
View booking



View Test Results

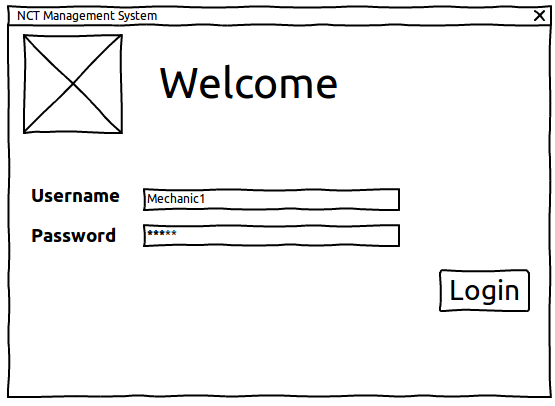


Database Design

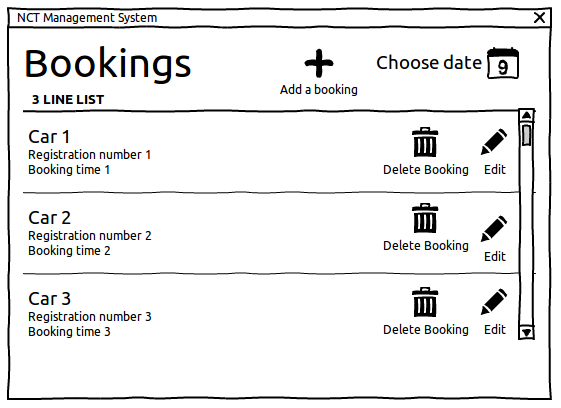


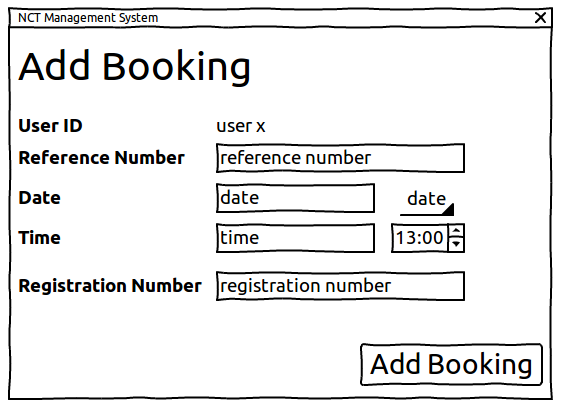
GUI Prototypes

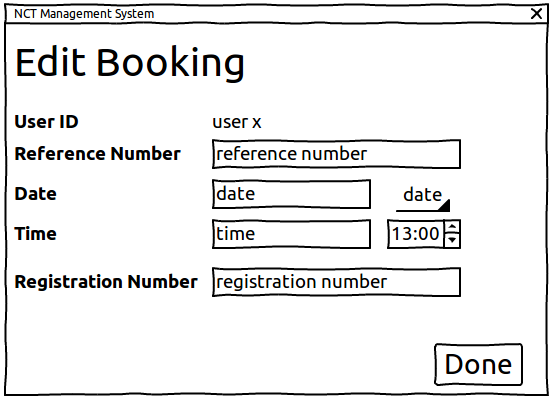
Desktop



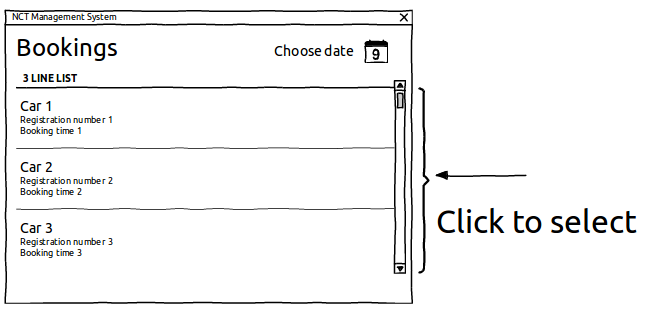
Admin

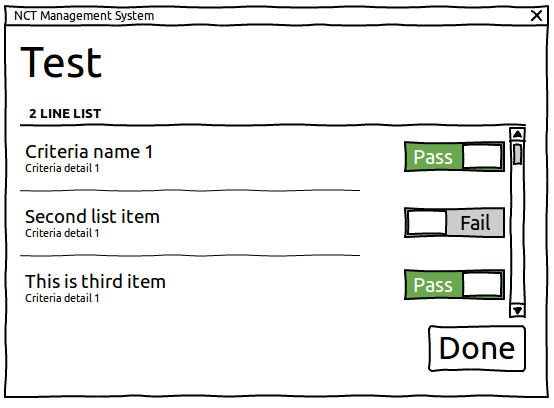




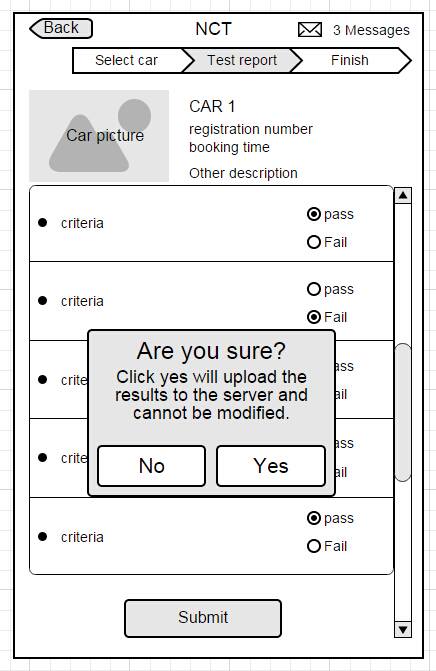
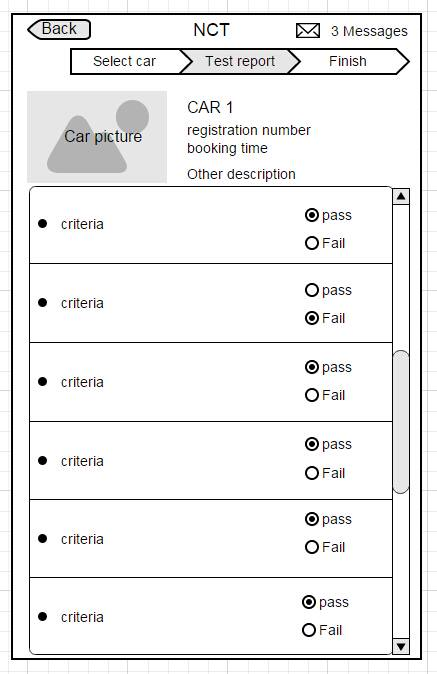
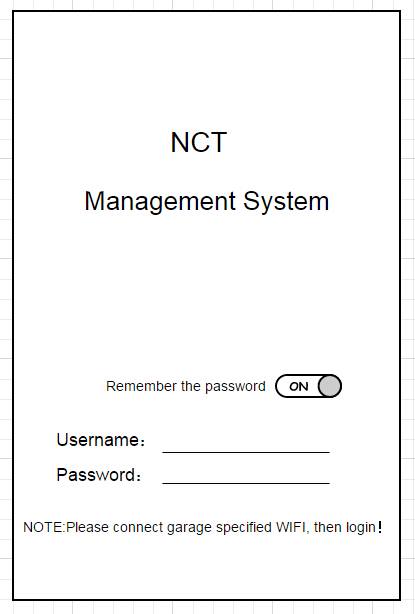
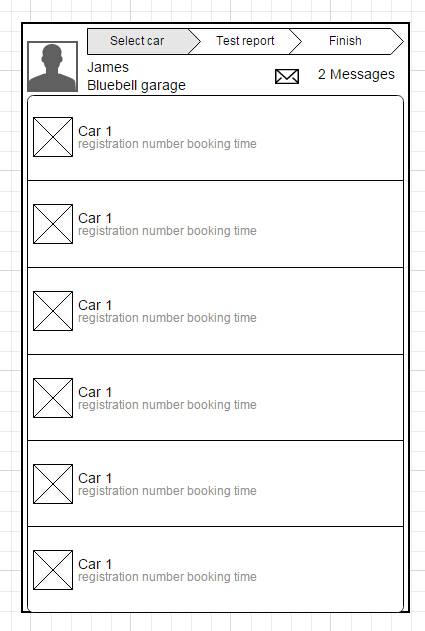


Mechanic





Android Application



##

