

SOFTWARE ENGINEERING

ASSIGNMENT PART 2

Solution Architecture Document

By Lean Bernardo, Harry Bishop, Daniel Brown,
Luke Butterick, Vince Verdadero

Table of Contents

SOFTWARE ENGINEERING	1
1. Introduction.....	3
1.1 Background	3
1.2 Purpose of the Document	3
1.3 Scope of the Document	3
1.4 Overview of the Document	4
2. Implementation view	5
2.1 Justification of packages and components that were identified.....	5
2.2 Implementation Diagram	6
2.3 Component Diagram	7
3. Use case model implications	8
3.1 Description of how the use case model has driven the design of the implementation and component views.....	8

1. Introduction

1.1 Background

COVID-19 has had an enormous effect all over the world on health, the economy, and industries. The UK Government has introduced the NHS Test and Trace Program to help them trace the spread of the coronavirus and isolate new infections.

1.2 Purpose of the Document

This is a Solution Architecture document for our company Successful Partners Software (SPSoft) which is tasked to create a Test and Trace software platform namely TTS.COVID-19 for the NHS. We aim to quickly reach potentially infected individuals to prevent the spread of the COVID 19 infection. The requirement is outlined in the project proposal section and extracted from <https://www.gov.uk/guidance/nhs-test-and-trace-how-it-works#contents>

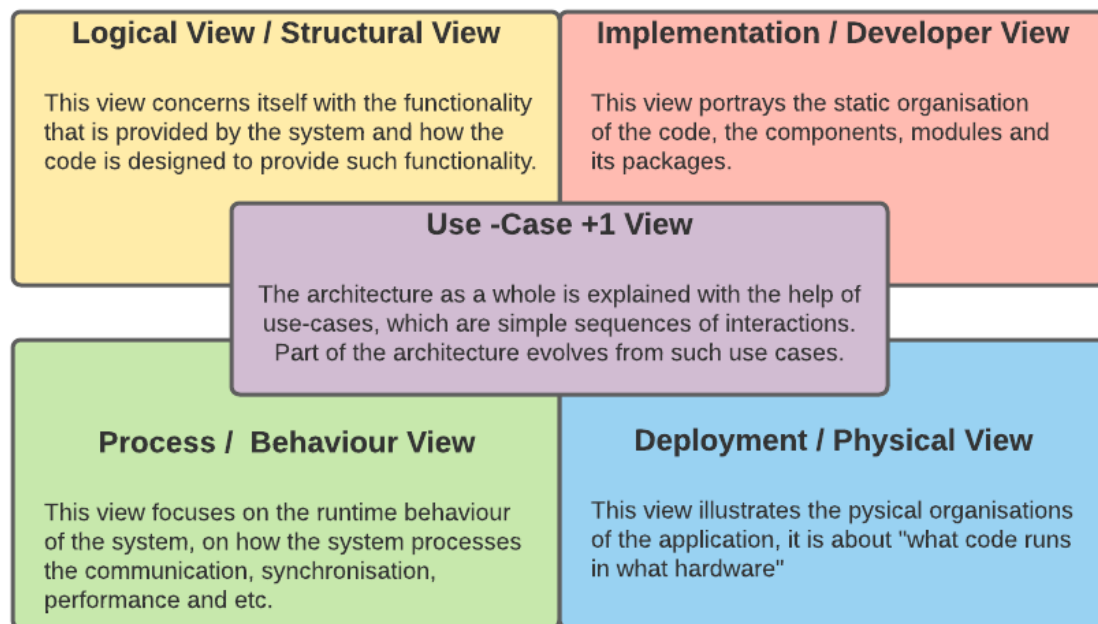
1.3 Scope of the Document

In this document, we will be talking about the architectural description of the structure, representing different elements of the TTS.COVID-19 System using the “4+1 views” models, to be able to describe the implementation and component view of the system's architecture. We will also show how a developed use-case model helps in assisting in the development of these views.

It aims to document and convey the system's major architectural decisions. It will also give us a more accurate representation or image of the system that we want to work on and all its involved components.

Our group have used Astah to create diagrams to produce this solution architecture document. As a group, we have initiated a use case view that

was identified in the requirement specification document. This will help us to visually see all the components of our made use-case and how are they all linked together and their functions that we will need to integrate into each one.



The above diagram is Philippe Krutchen's 4+1 view model, which proposes that there should be four fundamental architectural views, The logical view, the Process view, the Development view and the Physical view.

1.4 Overview of the Document

This document will be organised as follows:

Section 2. Implementation view

2.1 Justification of packages and components that were identified

2.2 Implementation Diagram

2.3 Component diagram

Section 3. Use case model implications

3.1 Description of how the use case model has driven the design of the implementation and component views

2. Implementation view

2.1 Justification of packages and components that were identified

Our task was to implement a Test and Trace Software platform for the NHS named TTS.COVID-19. It is software that would help in preventing the spread of the COVID-19.

We have identified several packages and components. There are two overarching components – the client and the server.

For the server, there are several key components. The most important of these is the Main Service, which is responsible for the communication of all other services. Since we are running a client-server model, this component is also responsible for the communication between itself and any clients that may connect to it. The system would be completely unusable without this component since it is responsible for intercommunication between all the different subsystems of the system as well as the storage and management of accounts.

The contact tracing service package is responsible for managing the different NHS contact tracers that act as a means for contacting anybody necessary. This includes external lab members, who are responsible for preparing and sending test kits to users who register to the app, as well as ordinary people who have been in contact with someone who tested positive for COVID-19.

The contact tracers also use the communication service component, which is made up of both the email and SMS service components. They are responsible for sending out emails and text messages to both lab members and people at risk of having COVID-19. Without these components, there would be no way for the contact tracers to get in contact with those who are needed. The use of a communication service component allows for centralised management of both the email and SMS service components, which is beneficial for the admins who use the service.

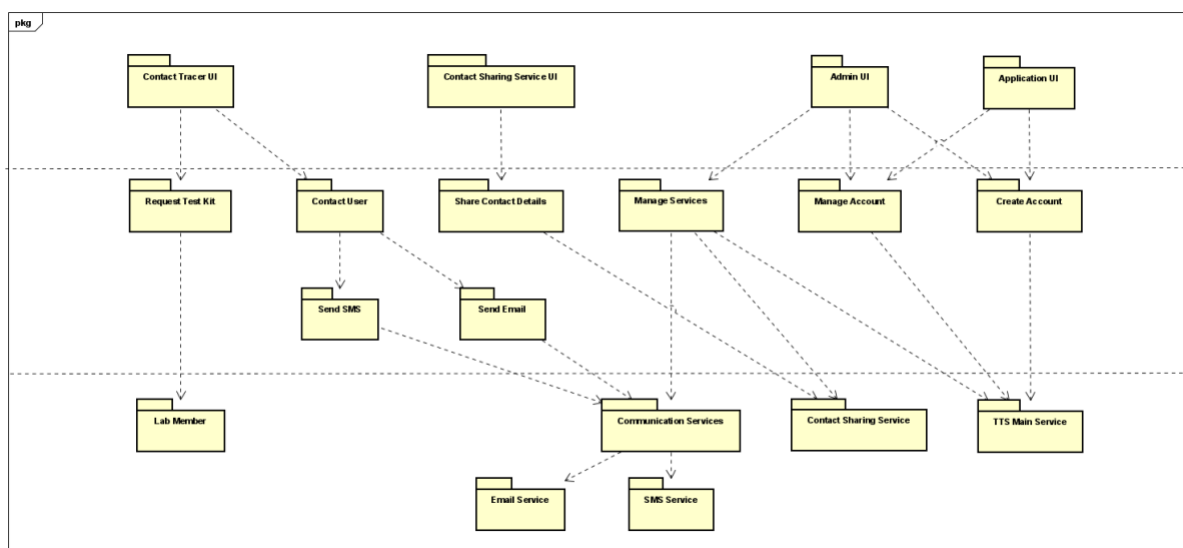
The packages related to the main service are the management and creation of accounts, which are controlled by both the application UI and admin UI. The admin and application UI are both client-side and are responsible for sending all the inputs and requests that an admin or regular user may make, respectively. These packages are important because

without them neither users nor admins would be able to interact with the backend of the system, rendering the whole app useless.

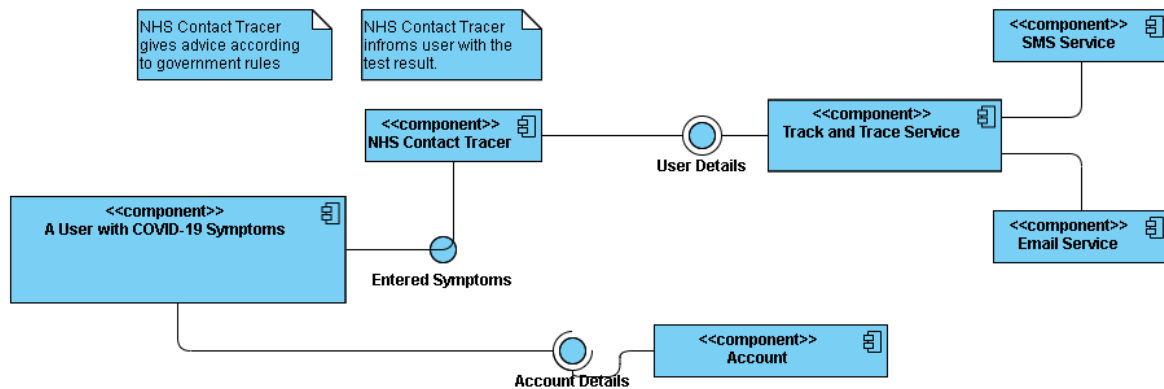
The admin UI is also responsible for the management of all other subsystems on the server. This is important because it allows for data that would otherwise be “stuck” on the backend to be accessible and managed securely, as well as giving the admin the ability to solve any possible issues that might occur during the usage of the server. It is different from the application UI because regular users cannot also be admins of the system to ensure the security of data.

The contact sharing UI is responsible for allowing users that are COVID-19 positive to share the contact details of those that they were in contact with. The service offers a more secure and direct way of sharing the details instead of just sending the data over text or email, which could potentially be intercepted and therefore jeopardize the data of other people. The data is then sent straight to the contact sharing service which manages all this contact data.

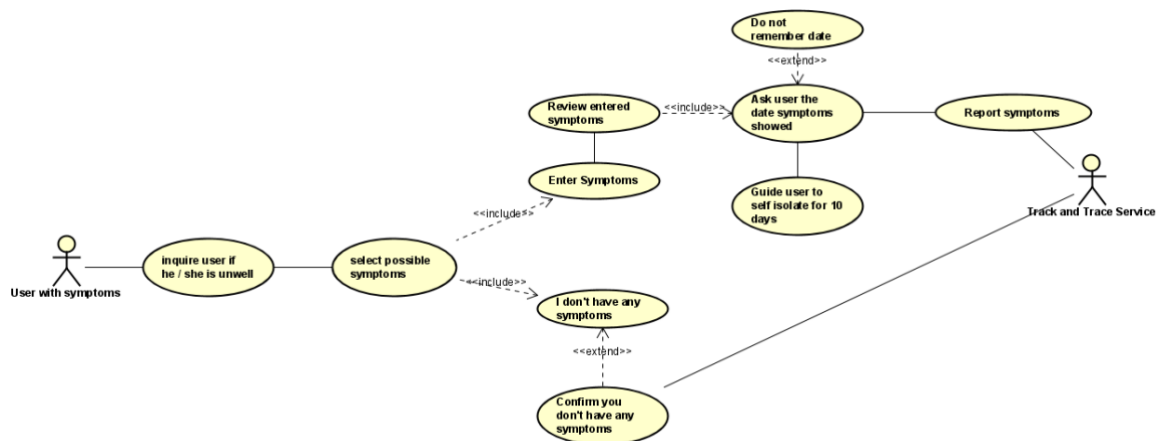
2.2 Implementation Diagram

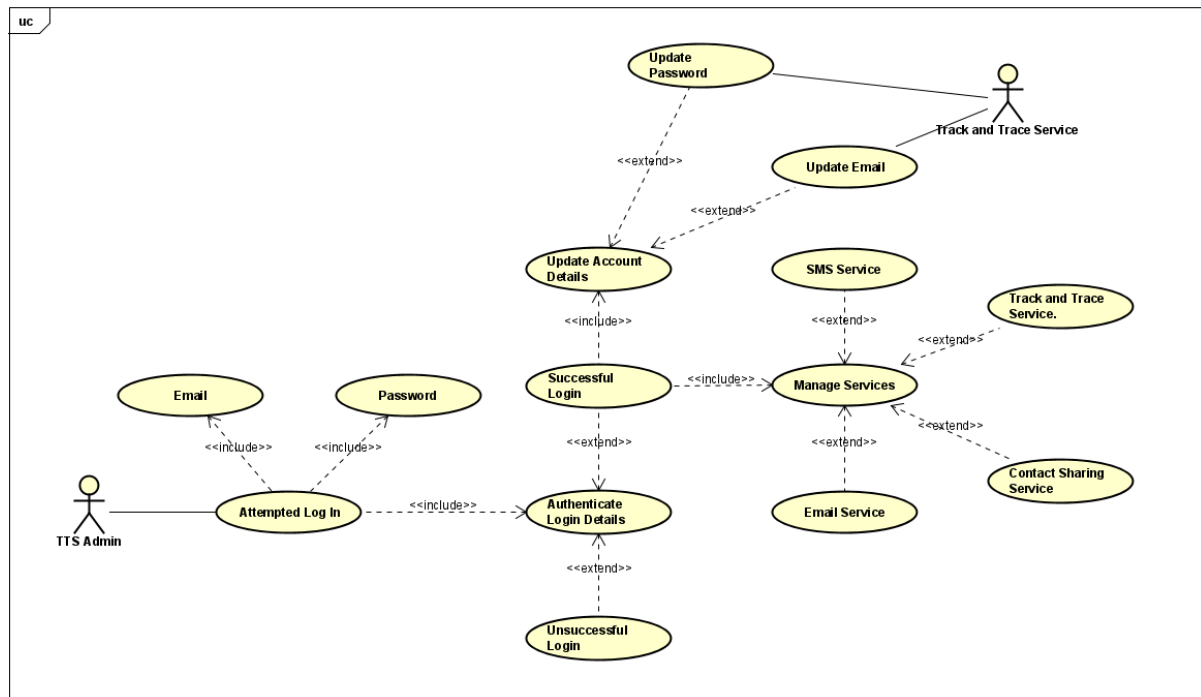


2.3 Component Diagram



Use-Case View





3. Use case model implications

3.1 Description of how the use case model has driven the design of the implementation and component views

Creating use-case views helped us to visually see how every component of our use-cases are linked to each other and their functionalities that are incorporated into each one of the use cases. It shows all the scenarios that represent the functionality and the significance.

With the help of the Use-case views, we were able to design the implementation and components views, that portrays all the organization of the code, the components and all its packages. It also helped us to go into more depth by being able to list the different interfaces and control components that were available. We were also able to include the services, and how they are connected to the system.

By creating the use-case views first, gave us the general idea of the system, on how we want the system to run. It gave us a more accurate

representation of what kind of system we are about to tackle on work on. We were able to code the prototype completely because we knew what links are needed to make the code in each of the use cases.