**Use Case Narratives:**

1. **Input values –** This is when a user chooses specific stations and times. It includes selecting the departure station and selecting the destination station. The user also has an option of selecting a departure time and/or arrival time.
2. **Calculate shortest route –** The inputted stations (and possibly times) are then passed into an algorithm which will find the shortest route between the stations.
3. **View timetable –** The user can also just simply view the times that trains arrive at specific stations.

**Typical course of events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User chooses departure station. | 1. Verifies station |
| 1. User chooses destination station | 1. Verifies station |
| 1. User chooses departure time (and/or arrival time) | 1. Calculates and returns shortest route |
|  |  |

**Alternative course of events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User chooses invalid departure station. | 1. Returns error and prompts valid station (back to step 1) |
| 1. User chooses times which are unrealistic | 1. Returns error of unachievable times and prompt user for realistic times (back to step 5) |
| 1. User chooses invalid destination station | 1. Returns error and prompts valid station (back to step 3) |
|  |  |

**Analysis Class Diagram**

There are only a few railway lines which all include several stations. Some of these stations may be apart of more than one line – this is where a transfer could occur. Each line also has a few trains which run daily on specific schedules. People (users) interact with the web interface (system) by inputting the required information, as well as optional additional information. The interface will then calculate the shortest route between two stations and retrieve a list of trains (with times) which achieve this shortest route