Initial Deliverable

Benjamin Jones

P2423891

Computing

Teacher Timetable

Adam Machowczyk, Eddie Osei-Poku

# Extended discussion of core functionality and product integration

The core functionality of my product is a teacher timetable. Admins can generate and manipulate a timetable for each teacher account, changing which room they are in per day for a month. I am doing it with the view of a single week to prevent issues about too wide of a timeframe being available in the software since this is a prototype and is not intended to be used outside of this module.

Admins will be the ones to create and manage teacher accounts, since it is intended to be a system used within a school where the public should not be able to make their own accounts. Accounts will still use appropriate security however, such as features to prevent SQL injections or XSS attacks, as well as keeping passwords in the database hashed. Inputs will be validated to make sure they don’t case issues later as well. Testing will be done on the validation to make sure bad values cannot get through.

Admins will be able to add, edit and delete rooms available on the timetable. Teachers will be able to edit room bookings on the timetable, to teach in a different room. Rooms available to be taught in will depend on what admins have added.

Teachers and admins will be able to message each other, intended for use as a way of teachers to request timetable changes or to report other issues. Admins will be able to select the teacher to send a message to, but teachers can only send messages to the single admin account since it is designed for support and not casual use.

My product integrates with the rest of the other team members products to create an overall system. This is done by sharing a database and some shared classes where functionality overlaps to reduce redundant code.

# Use Case Descriptions and Diagrams

|  |  |
| --- | --- |
| Use Case Name | Add room |
| Use Case Description | Admin adds a room to the list of available rooms along with the room’s relevant details |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin views the list of rooms  Admin clicks Add Room Button  Admin enters room details  Admin clicks Add Room Button |
| Alternate pathways | N/A |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem  Admin does not get correct login details |

|  |  |
| --- | --- |
| Use Case Name | Filter rooms |
| Use Case Description | Filter the results in the teacher list box |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin clicks manage rooms button  Admin enters room’s details  Admin presses search button |
| Alternate pathways | Admin logs in  Admin clicks manage rooms button  Admin presses show all button |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem |

|  |  |
| --- | --- |
| Use Case Name | View rooms |
| Use Case Description | Admin views a list of available rooms |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin views the list of rooms |
| Alternate pathways | N/A |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem  Admin does not get correct login details |

|  |  |
| --- | --- |
| Use Case Name | Edit room |
| Use Case Description | Admin edits an existing room from the list of available rooms |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin views the list of rooms  Admin clicks the room they want to edit  Admin enters room details  Admin clicks Save Room Details button |
| Alternate pathways | Admin logs in  Admin views the list of rooms  Admin applies a filter to the list of rooms  Admin clicks the room they want to edit  Admin enters room details  Admin clicks Save Room Details button |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem  Admin does not get correct login details |

|  |  |
| --- | --- |
| Use Case Name | Delete room |
| Use Case Description | Admin deletes an existing room from the list of available rooms |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin views the list of rooms  Admin clicks the room they want to delete  Admin asked for confirmation  Admin confirms |
| Alternate pathways | Admin logs in  Admin views the list of rooms  Admin applies a filter to the list of rooms  Admin clicks the room they want to delete  Admin asked for confirmation  Admin confirms |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem  Admin does not get correct login details |

|  |  |
| --- | --- |
| Use Case Name | Contact message |
| Use Case Description | User sends a contact message to another user |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin, teacher |
| Locations | On-line |
| Primary pathway | User logs in  User enters contact message  User sends message |
| Alternate pathways | N/A |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem  User does not get correct login details |

|  |  |
| --- | --- |
| Use Case Name | Edit room booking |
| Use Case Description | Teacher edits an existing room booking from their timetable |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Teacher |
| Locations | On-line |
| Primary pathway | Teacher logs in  Teacher views timetable  Teacher selects lesson slot to change room booking on  Teacher views rooms that are available (IT rooms)  Teacher selects room  Teacher saves information |
| Alternate pathways | Teacher logs in  Teacher views timetable  Teacher selects lesson slot to change room booking on  Teacher views rooms that are available  No rooms are available at that time |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem  User does not get correct login details |

|  |  |
| --- | --- |
| Use Case Name | View timetable |
| Use Case Description | Teacher views their timetable for the week |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Teacher |
| Locations | On-line |
| Primary pathway | Teacher logs in  Teacher views timetable |
| Alternate pathways | Teacher logs in  Teacher views timetable  Teacher views past weeks and future weeks |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem  User does not get correct login details |

|  |  |
| --- | --- |
| Use Case Name | Log in |
| Use Case Description | User logs in with their credentials |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Teacher, admin |
| Locations | On-line |
| Primary pathway | User logs in |
| Alternate pathways | User attempts log in  Details are incorrect and are returned to the login screen with an error message |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem |

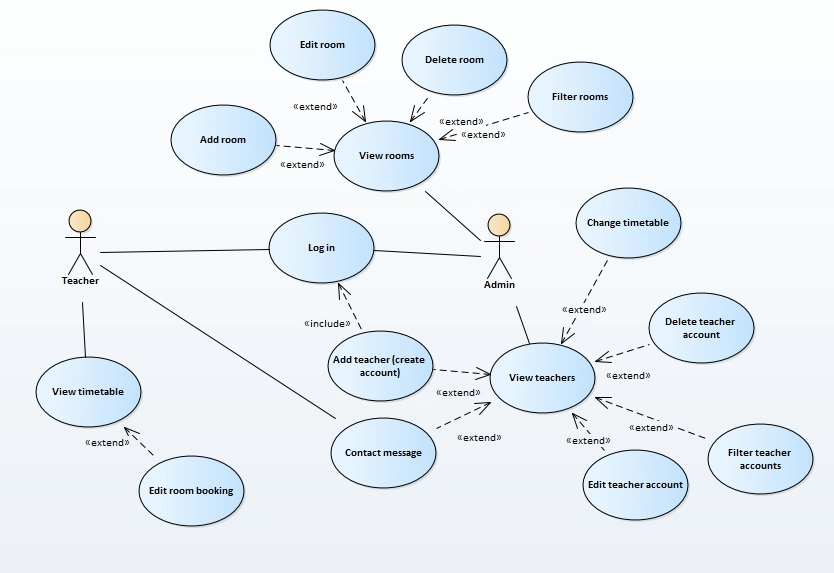
|  |  |
| --- | --- |
| Use Case Name | Create teacher account (add teacher) |
| Use Case Description | Admin creates a teacher account |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin clicks button to create new teacher account  Admin enters teacher’s details  Admin saves details and account is created |
| Alternate pathways | N/A |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem |

|  |  |
| --- | --- |
| Use Case Name | Filter teachers |
| Use Case Description | Filter the results in the teacher list box |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin clicks manage teacher button  Admin enters teacher’s details  Admin presses search button |
| Alternate pathways | Admin logs in  Admin clicks manage teacher button  Admin presses show all button |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem |

|  |  |
| --- | --- |
| Use Case Name | Edit teacher |
| Use Case Description | Admin edits details on a teacher’s account |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin views the list of teachers  Admin enters teacher’s details  Admin selects teacher and presses edit  Admin changes necessary details  Admin saves details and account is saved with new details |
| Alternate pathways | Admin logs in  Admin views the list of teachers  Admin enters teacher’s details  Admin selects teacher and presses edit  Admin presses cancel on edit page  Admin is taken back to teacher list page |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem |

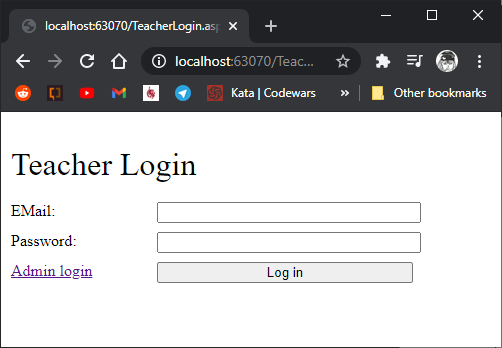
|  |  |
| --- | --- |
| Use Case Name | Delete teacher |
| Use Case Description | Admin deletes a teacher’s account |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin views the list of teachers  Admin enters teacher’s details  Admin selects teacher and presses delete  Admin confirms deletion  Admin is taken back to teacher list page |
| Alternate pathways | Admin logs in  Admin views the list of teachers  Admin enters teacher’s details  Admin selects teacher and presses delete  Admin cancels  Admin is taken back to teacher list page |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem |

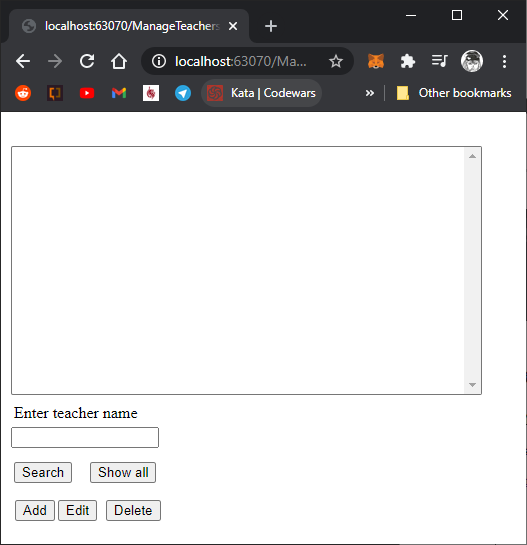
|  |  |
| --- | --- |
| Use Case Name | Edit timetable |
| Use Case Description | Admin generates timetable for a teacher |
| Use Case Author(s) | Benjamin Jones |
| Actor(s) | Admin |
| Locations | On-line |
| Primary pathway | Admin logs in  Admin clicks button to create new teacher account  Admin enters teacher’s details details  Admin saves details and account is created  Admin goes to timetable page  Admin enters timetable details  Admin confirms details |
| Alternate pathways | N/A |
| Exception pathways (What could possibly go wrong?) | Database connection fails  Error displayed to the user advising of connection problem |

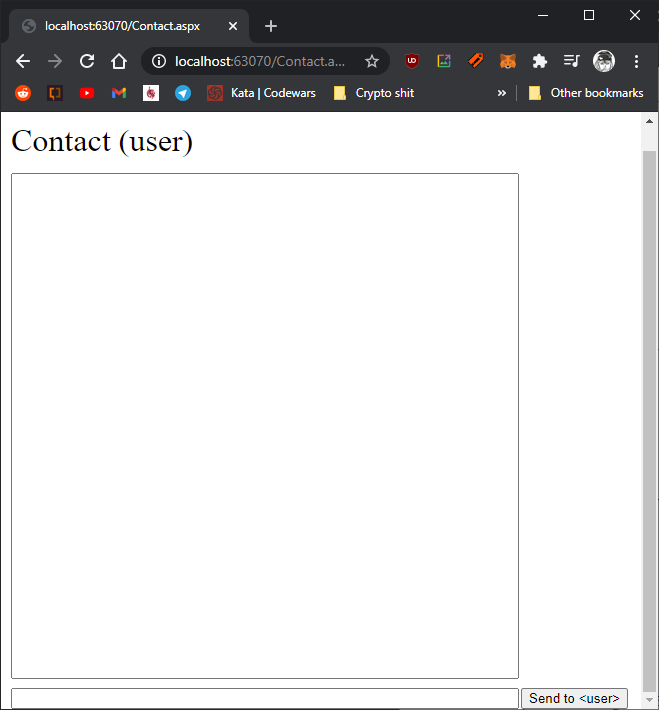


# Non-Functional Prototype

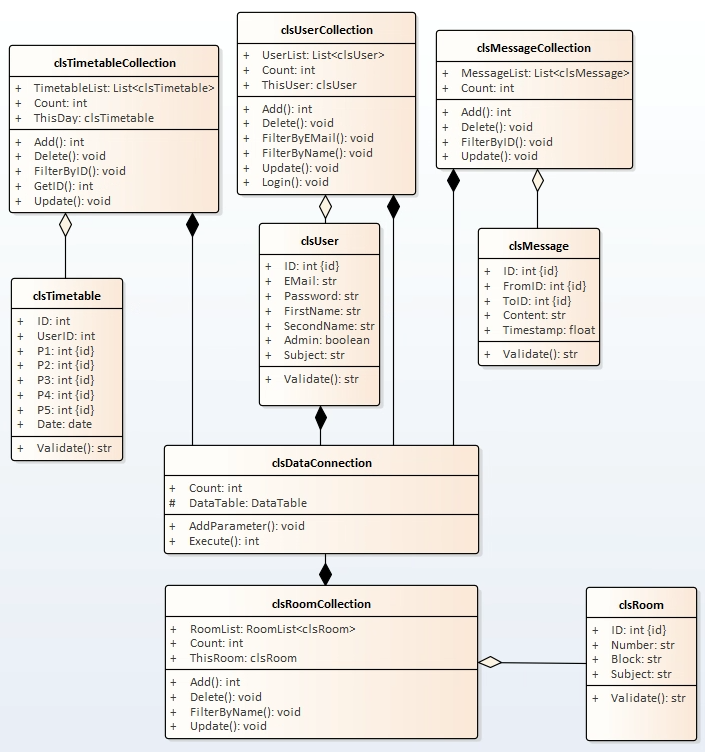
# 



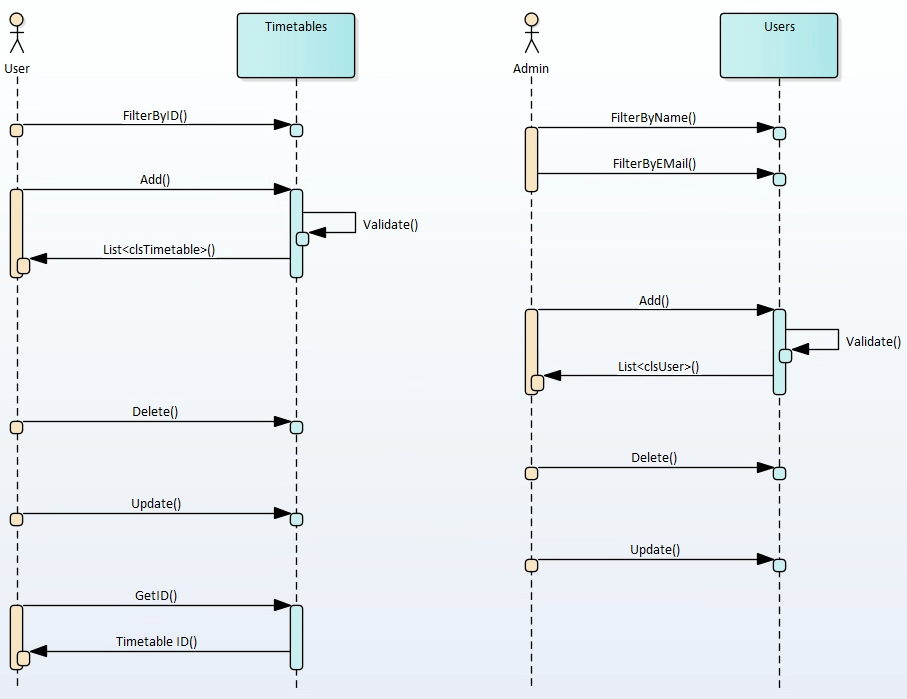


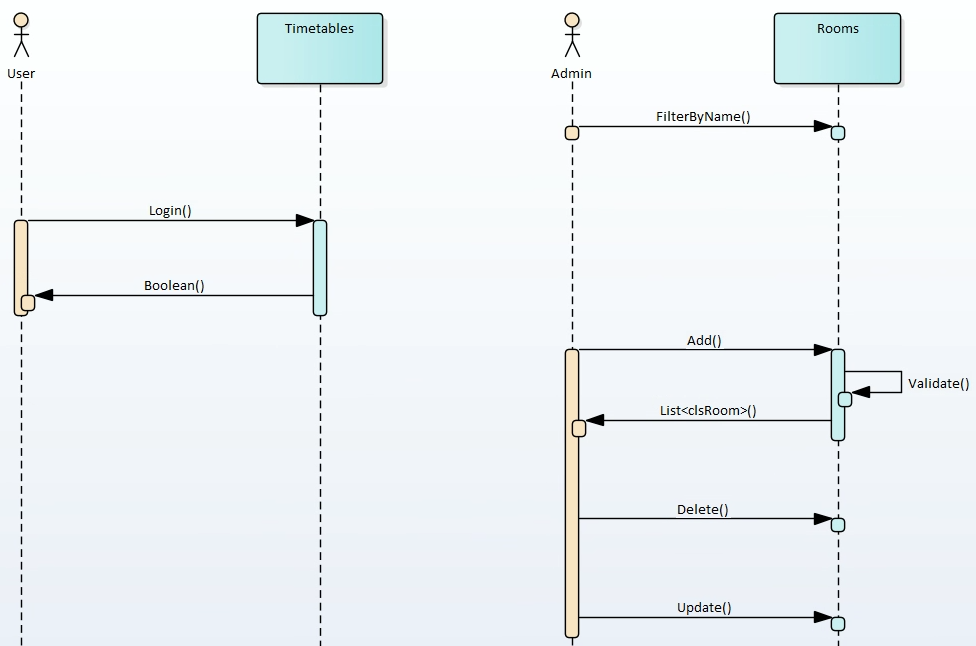


# Class Diagram

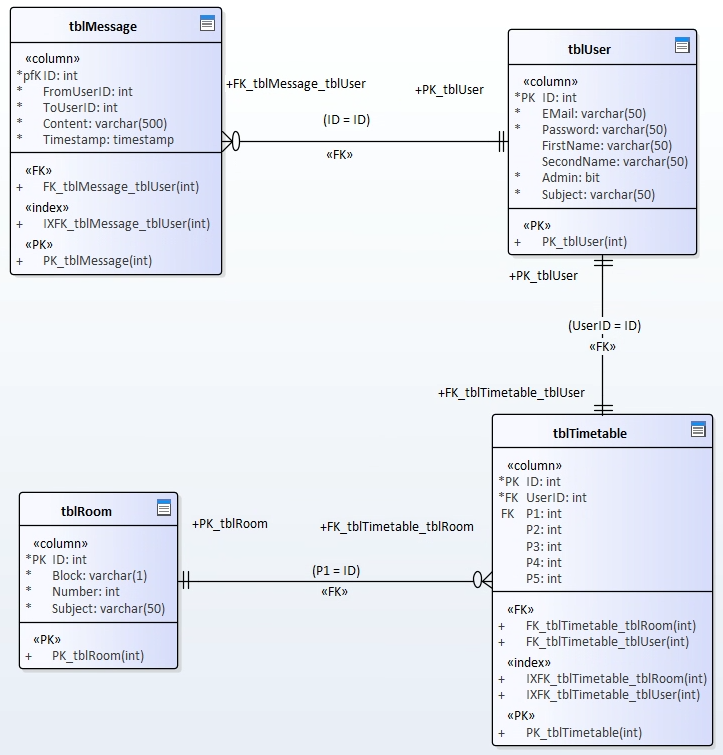


# Sequence Diagrams





# Entity Relationship Diagram



# Testing Strategy

My product will be tested in 2 different ways. Firstly, the code will need to be tested. This means making sure that different inputs, both expected and unexpected work without crashing the system and logging them to make sure the expected output is what actually happens. Values that are too high or entering a wrong data type could cause issues, so I will need to make sure these cases are handled with proper validation and tests to ensure this validation works as expected. To do this, I will create unit tests for each relevant class. These tests will need to run and all pass to be in a fully working state.

To make sure the product is usable, I will run a user trial. This will involve taking a willing test subject and asking them to do various tasks with the system to ensure it’s actually usable. The results of the test will be logged and analysed to make sure that any issues that arise get addressed before it is finished.

# Security Considerations

Despite the product not using real, sensitive data, security is still an important consideration. One important consideration is protecting passwords. All passwords will be securely hashed before storage using a safe and modern hashing algorithm. This is to prevent plaintext passwords from being leaked in the unlikely event of a database breach. Even though this is only a prototype product, it is still good security practice.

Other security features to consider is functionality to prevent SQL injections and XSS attacks. This means validating user input to filter out content that could be used maliciously, as well as just making sure all data that goes in is safe and will not cause crashes, intentionally or not.

# Code for Product

<https://github.com/benjimaestro/TeamDev2020/tree/benbranch>

# News report bibliography

Ask about finding sources and what is acceptable