**Part A**

R1: the software shall allow the user to login into the application this is so that the user can login into their unique account with an email address and a password

R2: the software shall allow the user to create an account, so each person has a unique account

R3: the software shall allow the user to log out of their account so the user can sign out of their account

R4: the software should have a forgot password feature so the user can recover their account.

R5: the software shall allow the user to enter payments for a specific time and cancel a particular payment history, this is so the user has some flexibility in recording their finances and keep track of finances which can then be used to produce expenses reports, and to cancel if they made a mistake.

R6: the software will provide a form for the entry of expenses; this is so that the user can clearly enter their expenses and enter the relevant information

R7: Allow the user to produce a finance report, this is because the application is so the user can keep record of their finances and see reports from different time periods

R8: Allow the user to create contacts and to keep records of those contacts that they transferred or received money from this is so the user can keep a history of who they are sending or receiving money from

NF1: The application will run on windows

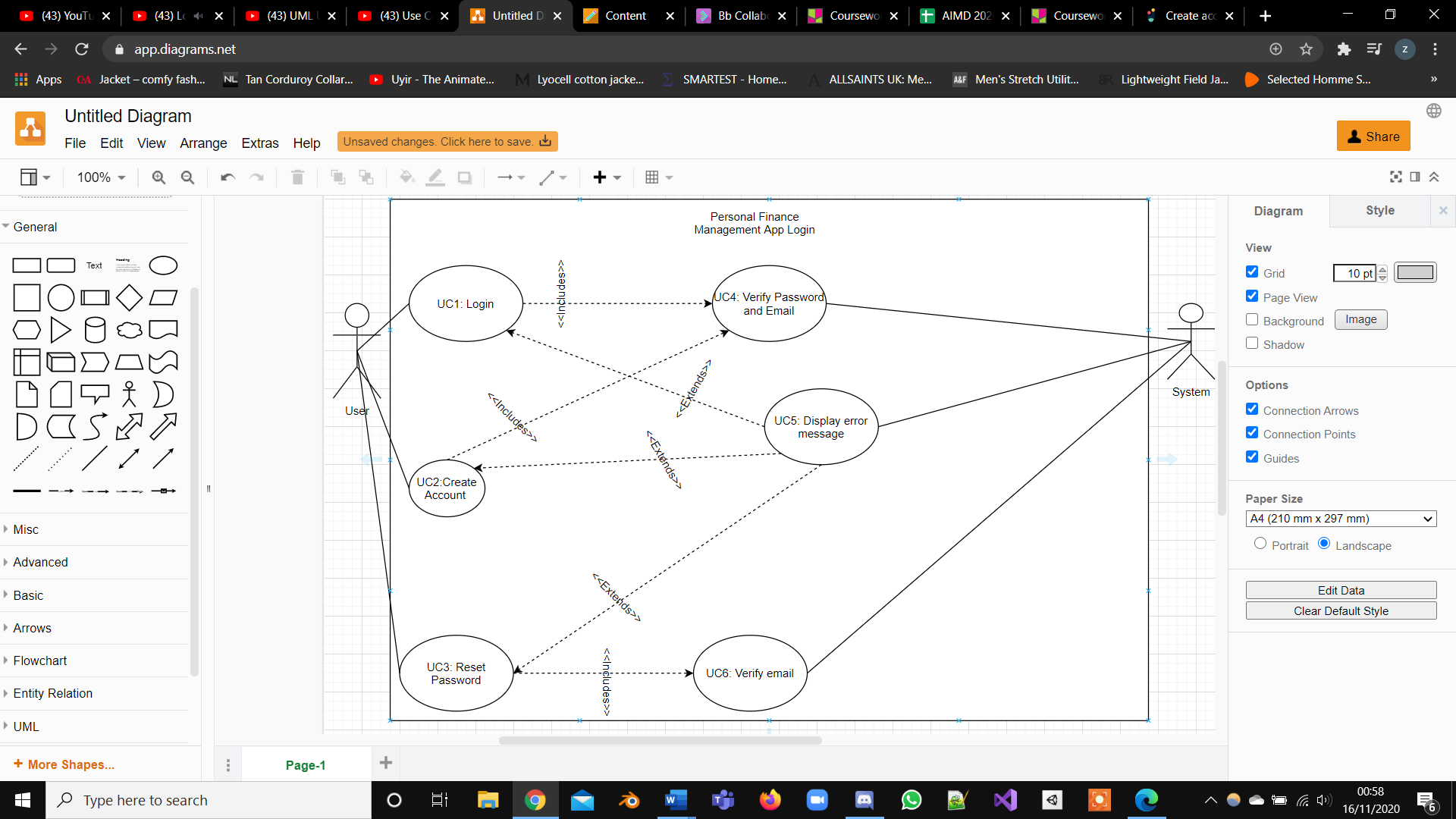
NF2: The application must be implemented using c# and windows forms.

NF3: The performance of the application will be at an optimal standard

NF4: The application will have few to no bugs

NF5: The application will be secured

**Part B**



|  |  |
| --- | --- |
| Use Case ID | UC:1 |
| Functional requirement  associated ID | FR 1  Login |
| Description | The user wanted to access to the system’s functions. |
| Level | Low Level Summary |
| Primary actor | User, System |
| Supporting Actors | - |
| Stakeholders and Interests | User: be able to access to the system’s functionalities.  System: give access to real users. |
| Pre-conditions | No user was logged in in the system. |
| Post-conditions | The user is allowed to access to the software’s functions. |
| Trigger | The user wishes to login to his/her account. |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. Starts the software. |  |
|  | 2. Opens a view displaying fields to fill  including Email and password |
| 3. Writes a Email |  |
| 4. Writes a password |  |
| 5. Selects the option to login. |  |
|  | 6. Checks the combination of Email and  password. |
|  |  |
|  | End of use case. |

|  |  |
| --- | --- |
| Use Case ID | UC:2 |
| Functional requirement  associated ID | FR 2  Create Account |
| Description | The user wanted to make an account with the application. |
| Level | Low Level Summary |
| Primary actor | User, System |
| Supporting Actors | - |
| Stakeholders and Interests | User: be able create an account with the application.  System: make an account for the user. |
| Pre-conditions | Email was not used before. |
| Post-conditions | The user can login and use the application. |
| Trigger | The user wishes to login to his/her account. |

|  |  |
| --- | --- |
| **User** | **System** |
| 1. Starts the software. |  |
|  | 2. Opens a view displaying fields to fill  including Email and password |
| 3. Writes a Email |  |
| 4. Writes a password |  |
| 5. Selects the option to create account. |  |
|  | 6. Checks that the Email and  Password matches the criteria. |
|  |  |
|  | End of use case. |

Main Success Scenario

|  |  |
| --- | --- |
| Use Case ID | UC:3 |
| Functional requirement  associated ID | FR 3  Reset Password |
| Description | The user wanted to make a new password. |
| Level | Low Level Summary |
| Primary actor | User, System |
| Supporting Actors | - |
| Stakeholders and Interests | User: be able create a new password with the application.  System: store the new password. |
| Pre-conditions | Email is stored on the system. |
| Post-conditions | The user can login and use the application with the new password. |
| Trigger | The user wishes to login to his/her account with the new password. |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. Starts the software. |  |
| 2. Click forgot Password |  |
|  | 3. Opens a view displaying field to fill  in Email |
| 4. Writes an Email |  |
|  | 5.Check if email is valid |
| 6. Writes a new password |  |
| 7. Selects the option to create a new password. |  |
|  | 8. Checks that the Password matches the criteria. |
|  |  |
|  | End of use case. |

|  |  |
| --- | --- |
| Use Case ID | UC:4 |
| Functional requirement  associated ID | FR 4  Verify password and email |
| Description | The application will check if the email and password meet the criteria and or exists on the system. |
| Level | Low Level Summary |
| Primary actor | System |
| Supporting Actors | - |
| Stakeholders and Interests | System: check if the email and password exists and or if the email and password matches the criteria. |
| Pre-conditions | Email and password entered in. |
| Post-conditions | The user can login or make an account. |
| Trigger | The user will login to their account or create an account. |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. Enters email and or password |  |
|  | 1. Password and email is checked to see if it exists |
|  | End of use case. |

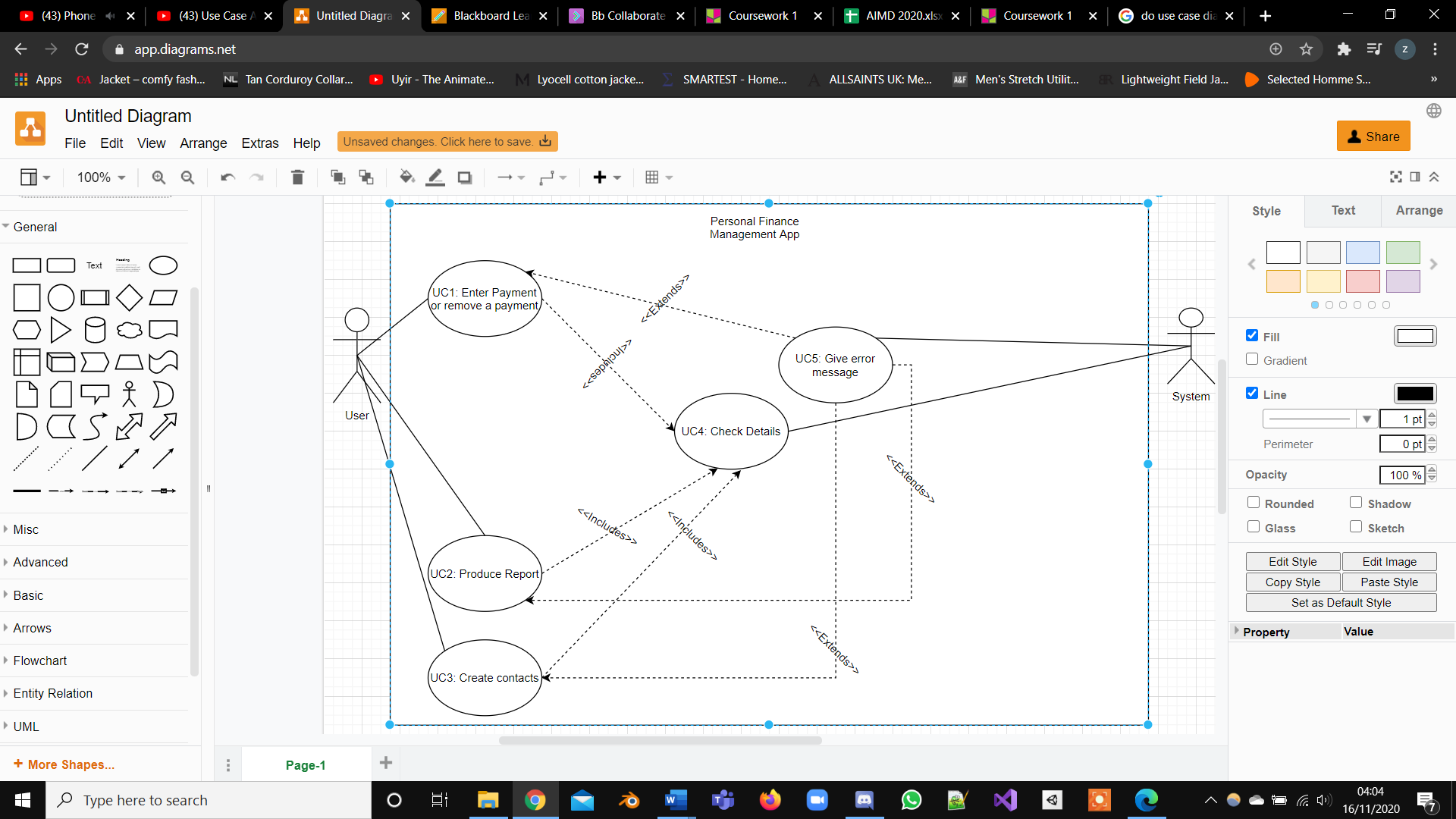
|  |  |
| --- | --- |
| Use Case ID | UC: 5 |
| Functional requirement  associated ID | FR 5  Display error message |
| Description | The application will display an error message if the user makes a mistake |
| Level | Low Level Summary |
| Primary actor | System |
| Supporting Actors | - |
| Stakeholders and Interests | System: prompt the user on their mistake. |
| Pre-conditions | Email and/or password entered in. |
| Post-conditions | Email address and or password is wrong or is not on the system. |
| Trigger | The user can login or make an account or reset their password. |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. Enters email and or password |  |
|  | 1. Display an error message explaining the error |
|  | End of use case. |

|  |  |
| --- | --- |
| Use Case ID | UC:6 |
| Functional requirement  associated ID | FR 6  Verify email |
| Description | The application will check if the email meet the criteria and or exists on the system. |
| Level | Low Level Summary |
| Primary actor | System |
| Supporting Actors | - |
| Stakeholders and Interests | System: check if the email exists and or if the email and password matches the criteria. |
| Pre-conditions | Email entered in. |
| Post-conditions | The user reset their password. |
| Trigger | The user will be able to reset their password. |

|  |  |
| --- | --- |
| **User** | **System** |
| 1. Enters email address |  |
|  | 1. email is checked to see if it exists |
|  | End of use case. |



|  |  |
| --- | --- |
| Use Case ID | UC:1 |
| Functional requirement  associated ID | FR 1  Enter Payment or delete payment |
| Description | The user wanted to make a payment log or remove a payment. |
| Level | Low Level Summary |
| Primary actor | User, System |
| Supporting Actors | - |
| Stakeholders and Interests | User: be able to log a payment or remove a payment.  System: store the payment history or remove the payment. |
| Pre-conditions | User is logged in. |
| Post-conditions | The user can create a payment log or remove the payment. |
| Trigger | The user wants to make a payment log or remove a payment log |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. selects make payment or remove payment from history. |  |
|  | 2. Opens field for them to enter finances and a date to make a payment log or search a payment |
| 3. Writes an amount of money |  |
| 4. Writes a date |  |
| 5. Selects the option to make a payment log or remove payment. |  |
|  | 6. Checks the date is real. |
|  |  |
|  | End of use case. |

|  |  |
| --- | --- |
| Use Case ID | UC:2 |
| Functional requirement  associated ID | FR 2  Produce report |
| Description | The user wanted to make a finance report. |
| Level | Low Level Summary |
| Primary actor | User, System |
| Supporting Actors | - |
| Stakeholders and Interests | User: be able to create a finance report.  System: store the finance report on the application. |
| Pre-conditions | User is logged in. |
| Post-conditions | The user can create a finance report. |
| Trigger | The user wants to make a finance report |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. selects make create finance report . |  |
|  | 2 finance report is created |
|  | End of use case. |

|  |  |
| --- | --- |
| Use Case ID | UC:3 |
| Functional requirement  associated ID | FR 3  Create contact |
| Description | The user wanted to make a contact that it receives money from or sends money to. |
| Level | Low Level Summary |
| Primary actor | User, System |
| Supporting Actors | - |
| Stakeholders and Interests | User: be able to make a contact of which the user receives money from or gives money to frequently.  System: store the contact on the application. |
| Pre-conditions | User is logged in. |
| Post-conditions | The user can create a contact. |
| Trigger | The user wants to make contact |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. selects make create contact. |  |
|  | 2. contact is created |
|  | End of use case. |

|  |  |
| --- | --- |
| Use Case ID | UC:4 |
| Functional requirement  associated ID | FR 4  Check details |
| Description | Checks the details of the payment log or contact spelling to see if it fits a criteria. |
| Level | Low Level Summary |
| Primary actor | System |
| Supporting Actors | - |
| Stakeholders and Interests | System: check if the payment log date is real and the contact name spelling fits a criteria |
| Pre-conditions | User is logged in. |
| Post-conditions | The user can make or delete a payment log or make a contact |
| Trigger | The user can make or delete a payment log or make a contact |

Main Success Scenario

|  |  |
| --- | --- |
| **User** | **System** |
| 1. enters date or enters a name for the contact |  |
|  | 2 check if date is real and name fits a criteria |
|  | End of use case. |

|  |  |
| --- | --- |
| Use Case ID | UC: 5 |
| Functional requirement  associated ID | FR 5  Display error message |
| Description | The application will display an error message if the user makes a mistake |
| Level | Low Level Summary |
| Primary actor | System |
| Supporting Actors | - |
| Stakeholders and Interests | System: prompt the user on their mistake. |
| Pre-conditions | Payment date or contact name entered |
| Post-conditions | Date is wrong or name does not meet the criteria |
| Trigger | The user can try again |

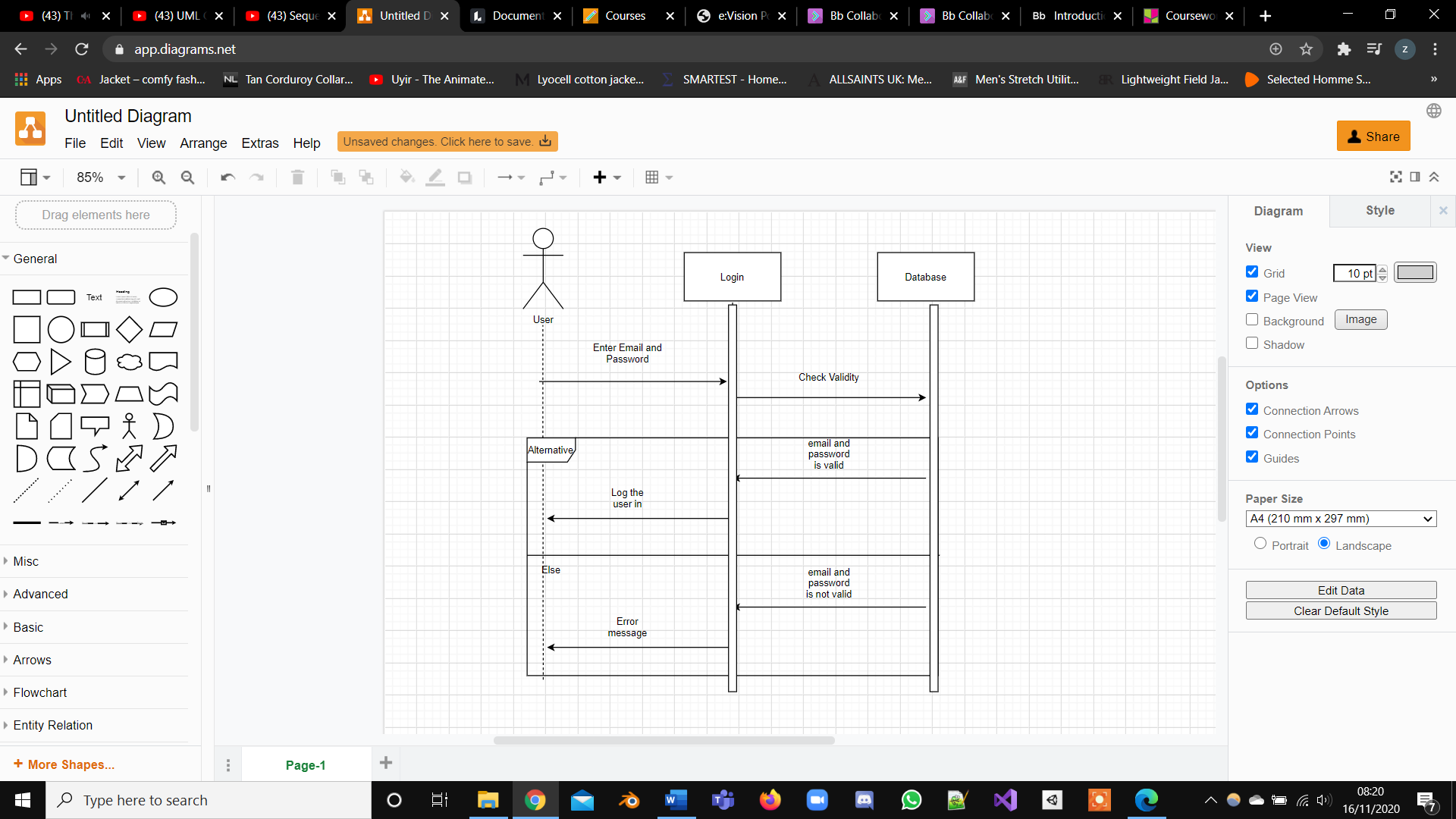
Main Success Scenario

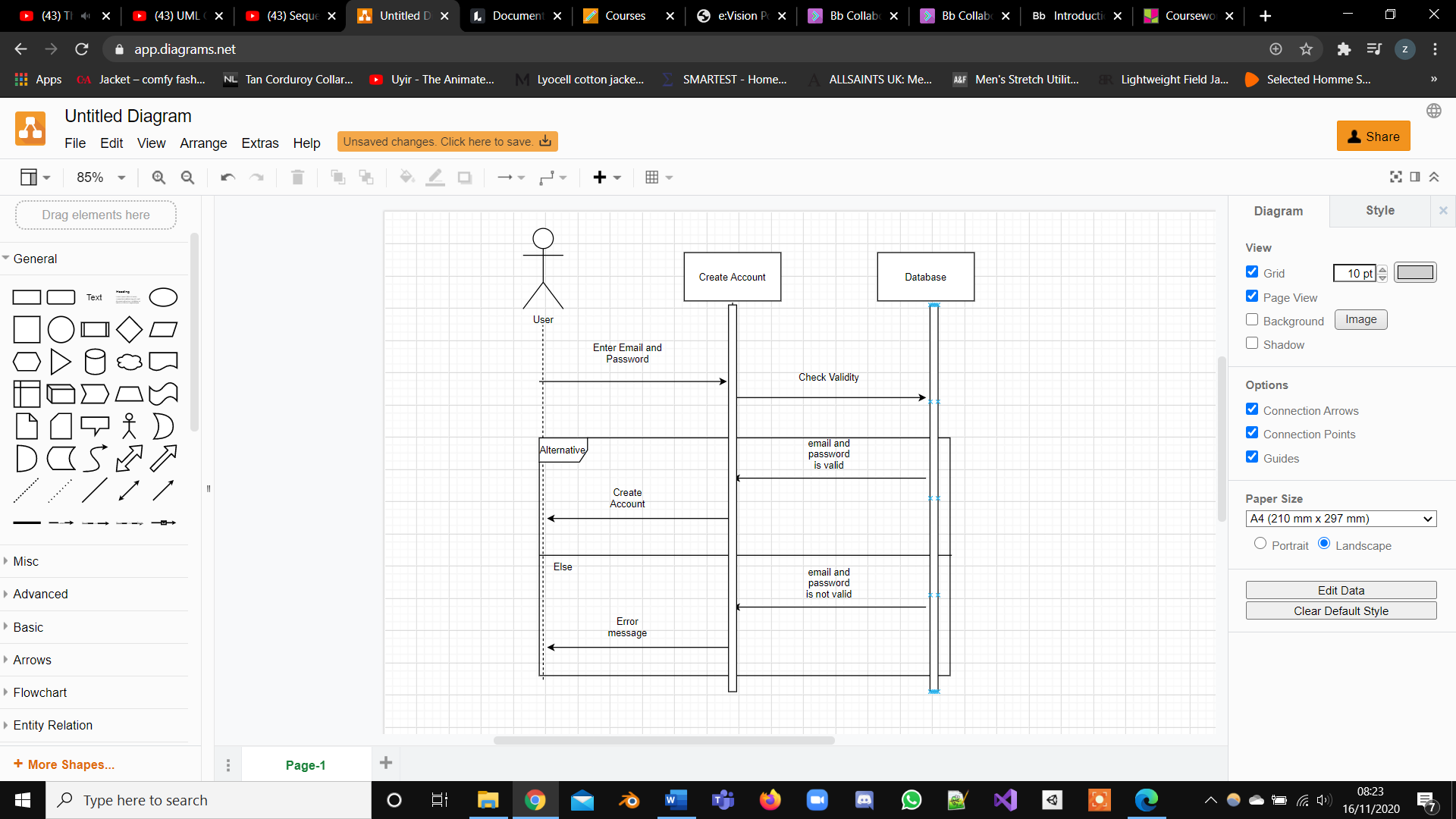
|  |  |
| --- | --- |
| **User** | **System** |
| 1. Enters date and or contact name |  |
|  | 1. Display an error message explaining the error |
|  | End of use case. |

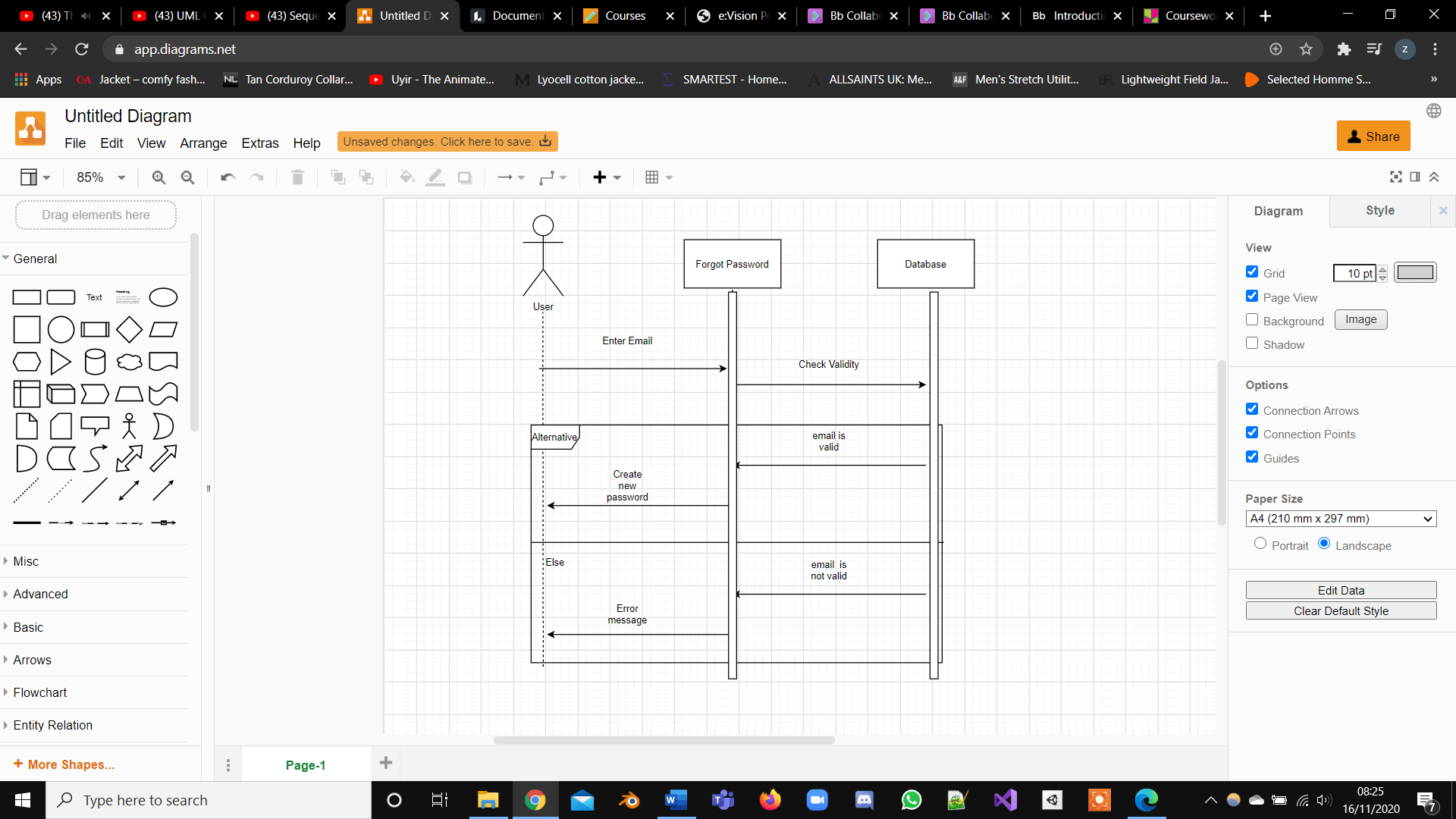
**Part C**

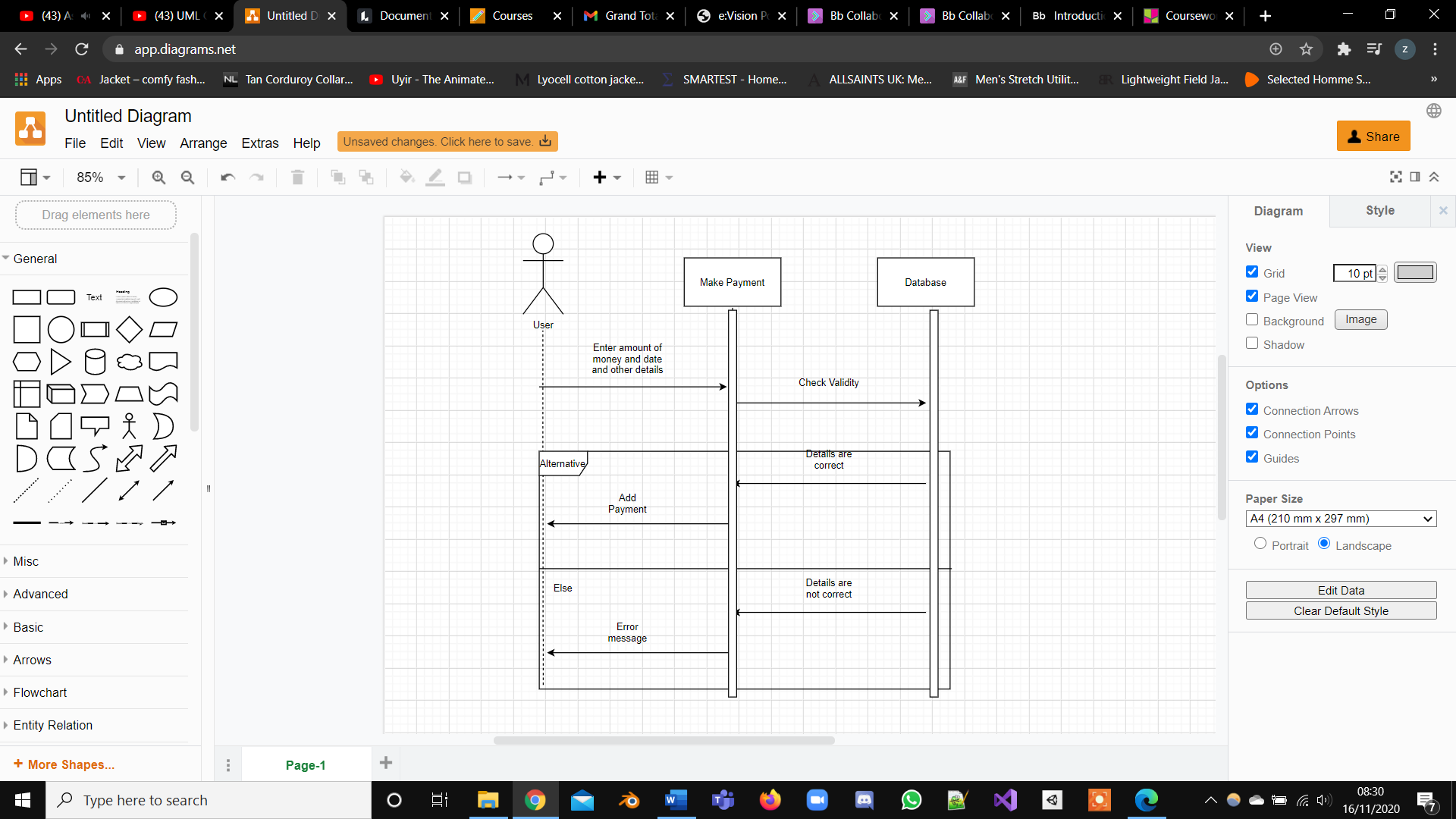
|  |  |  |  |
| --- | --- | --- | --- |
| Class Name | Type | Responsibility | Collaboration |
| Payment | model | It will hold information about payments such as the name, date and time | expenses |
| Login | view | Will allow the user to login to the account | Check\_Password  Check\_Email\_address |
| Forgot\_Password | view | Will allow the user to create a new password if they forget theirs | Check\_Email\_address |
| Create\_Account | view | Allow the user to create an account |  |
| Expenses | view | The user will make a payment log and record payments | Payment  Payment\_Report |
| Payment\_Report | view | Create a report for the payment | Payment |
| contact | view | Create a contact and to change details |  |

**Part D**

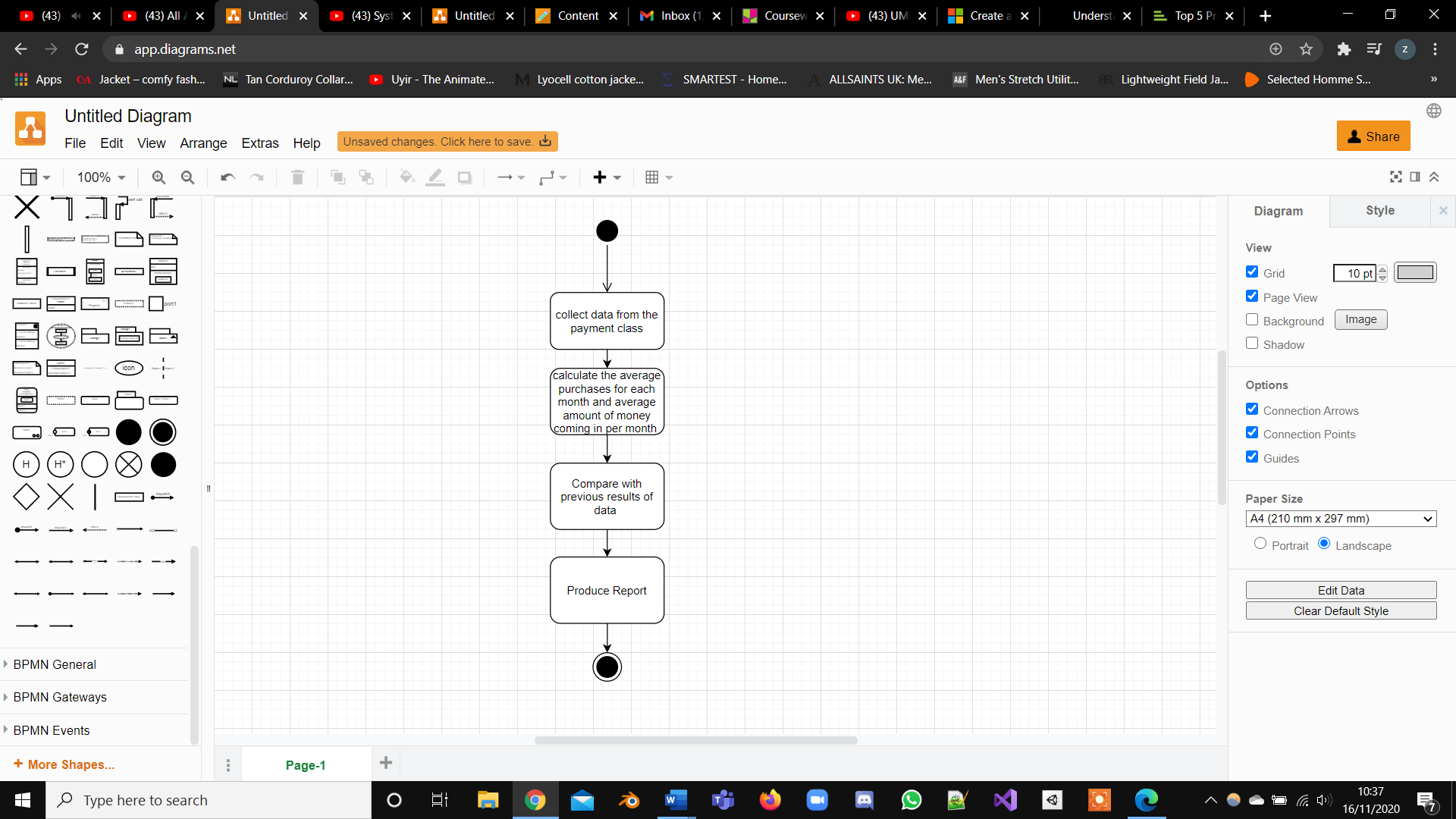








**Part E**



**Part F**

The overall project consisted of making multiple diagrams for a personal finance application, the functional requirements are explained thoroughly, and some non-functional requirements are listed but there could’ve have been more listed and explained as well to help give a better understanding what is needed, I could have also provided some drawings of the login page to help explain better the purpose of the functions.

The use case diagrams are kept concise with brief explanations and have a naming convention to. However some of the use cases could have been unnecessary such as making a use case for checking if the email exists, the use case descriptions explained each use case with two table, although a fail scenario table could have been added to show the user what will happen if something goes wrong and what should be done.

The CRC table is vague and does not go into depth about the functionalities and not enough scenarios are given to see so it was not a good use of the use case diagram, the domain model was not created as well this will be difficult for the programming aspect as some important parts are missing, but the types

Sequence diagrams were produced for some use case more should have been made, also they could have been explained in greater depth, so the user has a better understanding of what is happening. I should have also labelled the objects, so the user has a better understanding of their purpose

The activity diagram is rather simplistic it consists of taking averages however it is not explained in great depth this is bad because if a user tries to replicate it they will have a hard time doing so since it is unclear.