Nabeel Khan

CSC 4996/7

**Documentation of TODO Application**

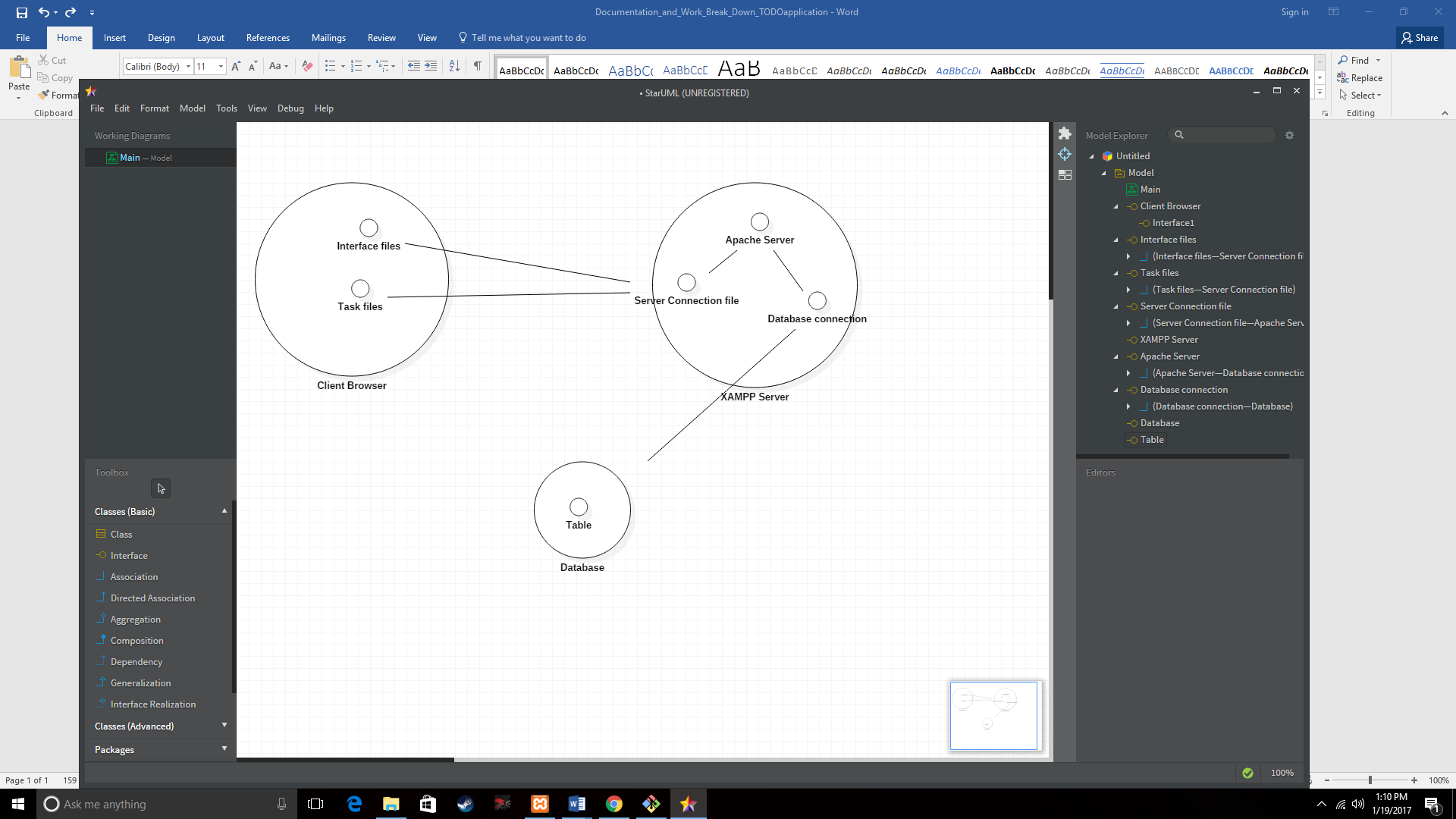
**Functional Requirements**

|  |  |
| --- | --- |
| Requirement | Requirement Necessity |
| An option to add a task will be present on the main page of the application, allowing users to type in and submit tasks to the database | Must |
| With the click of a button, the application will be able to fetch tasks from the database and display them to the user, in order of when the tasks were added. The table will show the tasks from earliest to latest. | Must |
| The application will be able to delete tasks by fetching the task table from the database and presenting the user with their choice on which tasks they want deleted | Must |

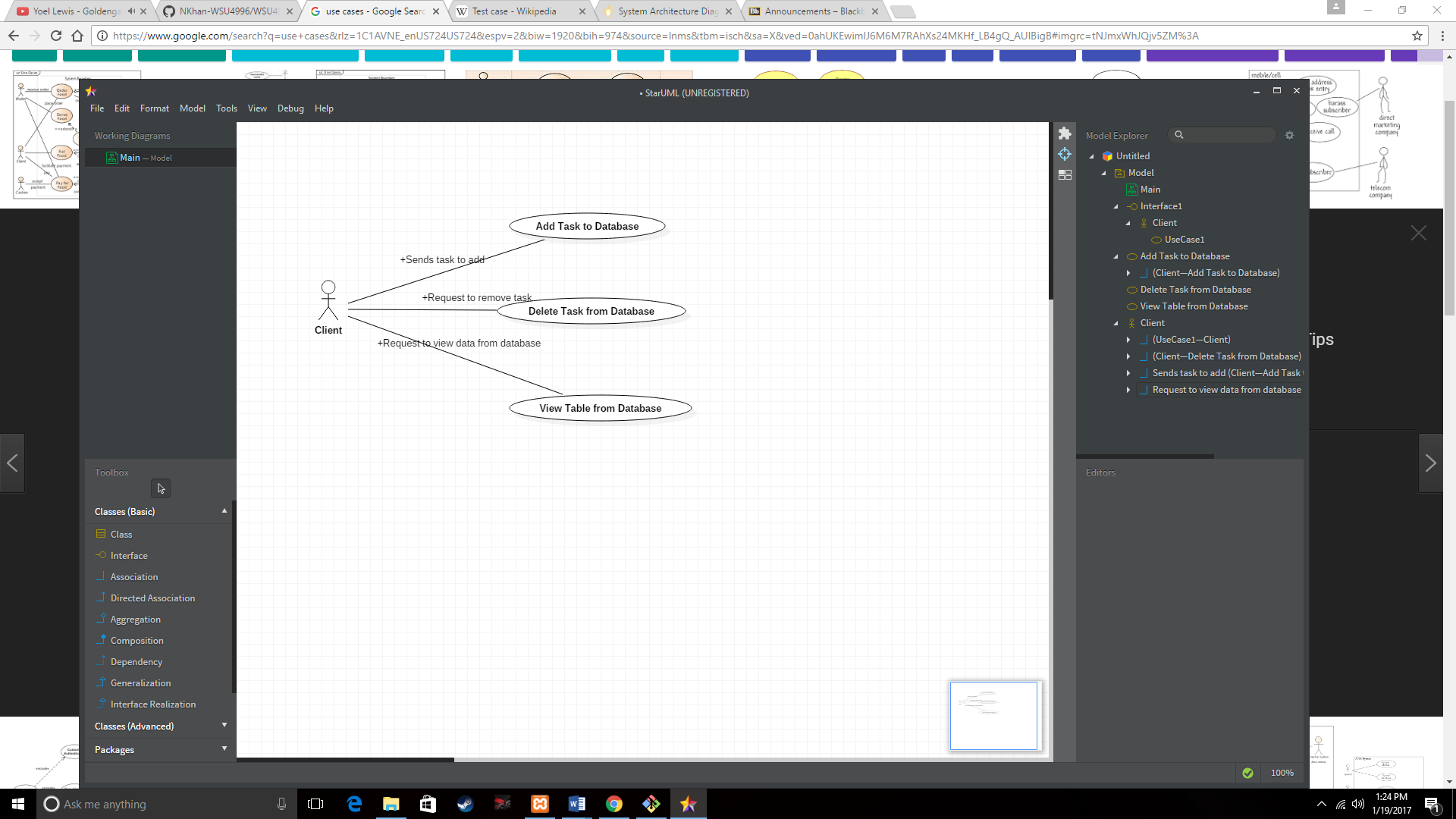
**Non-Functional Requirements**

|  |  |
| --- | --- |
| Requirement | Requirement Necessity |
| The application’s database will be easy to maintain, and easy to back up | Must |
| The application will be quick, and use very little time to load between pages | Want |
| The database will be safe from unwanted changes | Must |

**System Architecture Diagram**

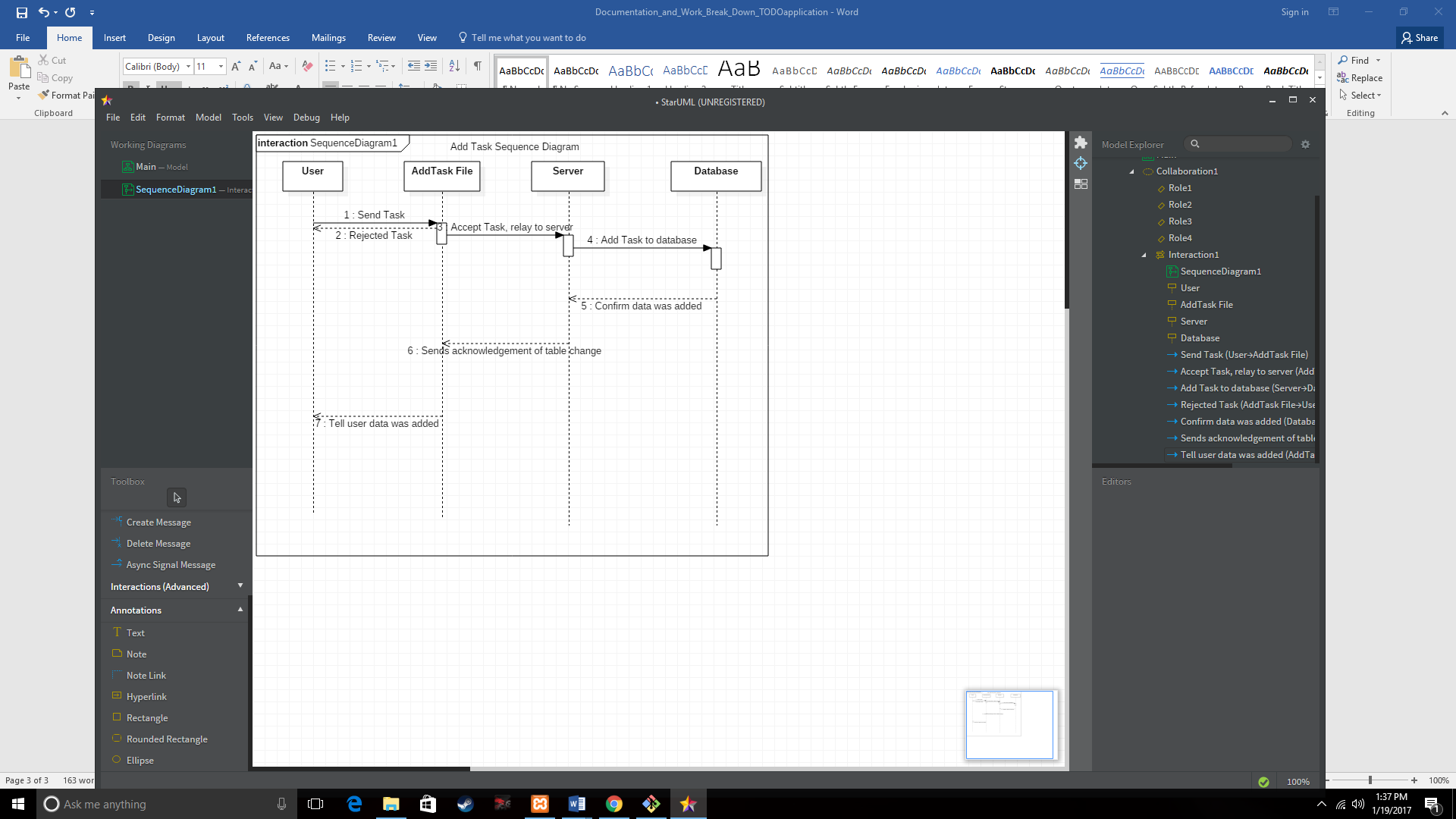


**Use Cases**

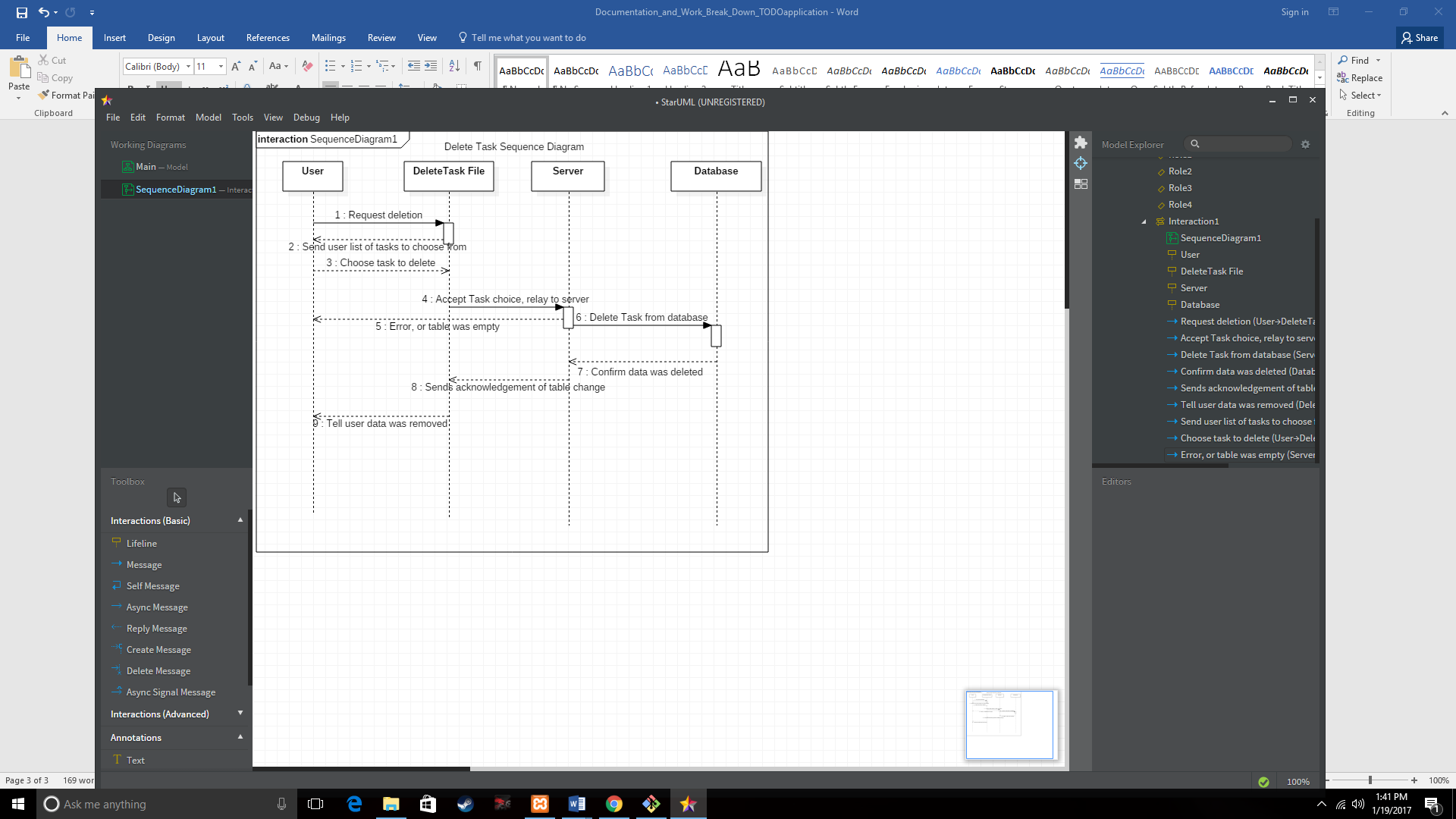


**Sequence Diagrams**

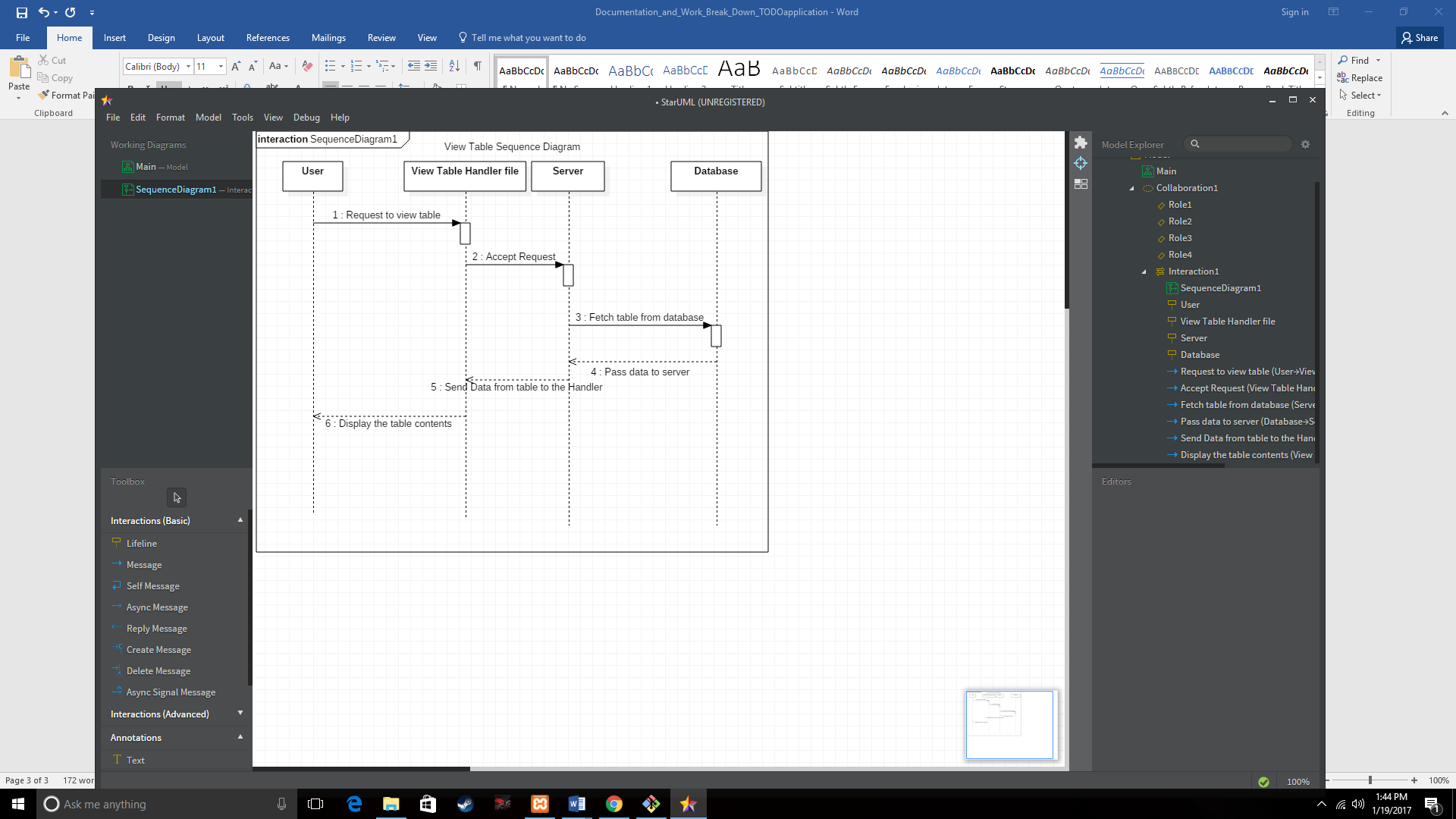
Add task diagram



Delete Task Diagram



View Tasks Diagram



**Database Design** – Only one table was created

|  |  |  |
| --- | --- | --- |
|  | Tasks |  |
| Task | VARCHAR(100) |  |
| Date\_Added | TIMESTAMP |  |
| Task\_id | **MEDIUMINT** | **Primary Key** |

**Test Cases**

1. Type a task into the Add Task bar, then click the “Add Task” button. (Check Database to confirm task was added). Repeat until sizeable table is created.
2. Click the view button to view the table, ensuring all tasks that were added in previous step are present in the table
3. Press the delete button. Ensure all tasks in list are on the drop down, then select a task to be deleted.

**Work Break Down**

Task – Create an application that allows a user to manage a to-do list by simply adding, viewing, and deleting tasks. The application will involve client-server methods using a database.

Estimated Number of Hours to Complete – 40 hours

Actual Time Spent to Complete – 22 hours

Notes: Part of the shorter completion time was due to an over-estimation in how long it would take to learn the technologies necessary to design this application, such as PHP and MySQL.

Issues: Due to a self-imposed lack of time, creating two other tables to expand upon the current application, such as a user log-in credentials table, was forgone in the interest of having the mandatory requirements fully developed first.