Appendix B

Project Set Up Instruction

Github Link: https://github.com/skrishan7/Thesis-Collaborative-UML-Modelling-Tool

B.1 Initial Set Up

1. Download Node.js and follow the installation wizard.
2. Download MongoDB Community Edition and follow the installation wizard.
3. Open Task Manager and go to Services tab and located MongoDB. Right click to start service.
4. To set up a local MongoDB database in command interpreter, type the following commands

   ```
   cd C:\
   md "\data\db"
   "C:\Program Files\MongoDB\Server\4.2\bin\mongod.exe" --dbpath="c:\data\db"
   "C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe"
   ```

5. Clone project from Github Link above.
6. The pusher is set up to my account while the application will still be functional as long as my pusher account exists. However, in order to be able to see and debug requests being sent to pusher, it is recommended you set up your own account. Create a new account on Pusher and log in. In the Channels Tab, click create a 'New App' and follow the wizard. Go to your app and find the 'App Keys' details.
7. In Server folder of the project, locate *app.js* file and update pusher details with your app keys on *line 13*.
8. Go to command prompt and type

   ```
   ipconfig
   ```

   to identify your IP address. Replace *'192.168.0.17'* with your IP address on *line 24* in *app.js* for MongoDB connection. Updating this IP address will let you collaborate on multiple machines on your network. This address needs to be updated every time your IP address changes.
9. The IP address also needs to be updated in the following files Client folder:

- protractor.conf.js : baseUrl parameter on line 19
- pusher.service.ts : baseUrl parameter on line 8
- uml.service.ts : baseUrl parameter on line 8

B.2 Starting the application

1. In first command prompt, go to *SERVER* folder and type the following command to install dependencies..

   ```
   ./server> npm install
   ```

2. In second command prompt, go to *CLIENT* folder and type the following command to install dependencies.

   ```
   ./client> npm install
   ```

3. To start the server in first command prompt:

   ```
   ./server> npx nodemon
   ```

4. To start the client in first command prompt:

   ```
   ./client> npm start
   ```

5. The project can now be accessed at this URL:

   ```
   http://< IP address >:4200
   ```

Appendix C

API Requests Documentation

Path	/api/umls
Description	Get all UML documents in the database
Method	GET
Type	
application/json	

TABLE C.1: GET Request: /api/umls

Path	/api/uml/:filename
Description	Get UML documents in the database by filename
Method	GET
Type	
application/json	
Path Parameters	
filename	Required: true Type: string Description: The filename of UML document

TABLE C.2: GET Request: /api/uml/:filename

Path	/api/uml/id/:id
Description	Get UML document based from database using document ID (used in the scenario of loading an existing UML diagram from a link)
Method	GET
Produces	
application/json	
Path Parameters	
id	Required: true Type: string Description: The UML document ID

TABLE C.3: GET Request: /api/uml/id/:id

Path	/api/uml/:filename
Description	Update an existing UML document in the database
Method	PUT
Type	
application/json	
Path Parameters	
filename	Required: true Type: string Description: The filename of UML document
Body Example	

uml	`{`
	`"filename": "file1",`
	`"content": "@startuml ....@enduml",`
	`"encoded": "XYZ",`
	`"lastEditedBy": "editor1",`
	`"__v": 0`
	`}`

TABLE C.4: PUT Request: /api/uml/:filename

Path	/api/uml/:filename
Description	Delete an existing UML document in the database
Method	DELETE
Produces	
application/json	
Path Parameters	
filename	Required: true Type: string Description: The filename of UML document

TABLE C.5: DELETE Request: /api/uml/:filename

Path	/api/uml
Description	Create a new UML document in the database
Method	POST
Type	
application/json	
Body Example	

uml	{
	"filename": "file1",
	"content": "@startuml\r\n\r\ntitle Activity Diagram
	\\n\r\n\r\nstart\r\n\r\n:Eat Hot Wings;\r\n\r\nnote
	left\r\n This is a Note...\r\n * Activity
	diagrams can begin with a Start\r\n * An activity
	is colon, some words, and a semicolon\r\n
	* Activity diagrams can end with a stop\r\nend
	note\r\n\r\n:Drink Homebrew;\r\n\r\nstop\r\n\r\n
	@enduml",
	"encoded": "VP2n3i8m34HtVuLdLFK3w5Aa8Z6nWCJ2ahQcj
	MWSb1Wg_XutWDY8iiHvTvTBdTH5lVi9G5admasd_61zOiDkYC
	xZMGOo1B3UEiLTK3on3Aa2aA244rqKqDPnv8Is7UvjNfPbdgv
	-bltRCc7d15iQM71c7Krmo04VND1z5URMXIb8WIbumZd4FXNe
	f3TD8i_vycyC6Vi9IHfki9oNPrkR9h9ShO-dDj9SBJKDWDeG
	-vON",
	"lastEditedBy": "editor1",
	"__v": 0
	}

TABLE C.6: POST Request: /api/uml

Path	/pusherevent
Description	Send a pusher event
Method	POST
Type	
application/json	
Body Example	

uml	{
	"encoded": "XYZ",
	"userid": "1212",
	}

TABLE C.7: POST Request: /pusherevent

Path	/api/codegen/:lang/:id
Description	Get all UML documents in the database by document id
Method	GET
Type	
application/json	
Path Parameters	
lang	Required: true Type: string Description: The language the code generation is required in e.g. Java
id	Required: true Type: string Description: The UML document ID

TABLE C.8: GET Request: /api/codegen/:lang/:id