

# GRACE DESIGN DOCUMENT

|                                    |   |
|------------------------------------|---|
| CRC Cards                          | 1 |
| CRC Model                          | 5 |
| System Interaction and Environment | 7 |
| System Architecture                | 7 |
| System Decomposition               | 8 |
| Prototype Design Mock-up           | 9 |

## ① WebApp

---

**Parent Class:** None

**Subclasses:** None

**Responsibilities**

- Authenticate user, upon filling sign in form
- Create account, upon filling register form
- Use the navigator to control application navigation

**Collaborators**

- Navigator

## ② File

---

**Parent Class:** None

**Subclasses:** None

**Responsibilities**

- Contains data
- Contains a unique file ID
- Has an list of users that can access the file
- Identifies its owner

**Collaborators**

- FileDAOImp

## ③ FileDAO <<Interface>>

---

**Parent Class:** None

**Subclasses:** None

**Responsibilities**

- Provide a layer of abstraction for file related database access

**Collaborators**

- None

#### ④ FileDAOImp

---

**Parent Class:** None

**Subclasses:** None

**Responsibilities**

- Collects files in file system
- Manages extant files in file system
- Adds new files in file system
- Deletes files in file system

**Collaborators**

- FileDAO (implements)

#### ⑤ User

---

**Parent Class:** None

**Subclasses:** None

**Responsibilities**

- Manage user account access information
- Owns entries in the file system
- Administers shared use for owned files

**Collaborators**

- UserDAOImp
- FileSystem
- UserPreferences
- Navigator

#### ⑥ UserDAO <<Interface>>

---

**Parent Class:** None

**Subclasses:** None

**Responsibilities**

- Provide a layer of abstraction for user related database access

**Collaborators**

- None

#### ⑦ UserDAOImp

---

**Parent Class:** None

**Subclasses:** None

**Responsibilities**

- Collects registered users
- Facilitates user management and existence control
- Allows for updating a user's UserPreferences

**Collaborators**

- UserDAO (implements)

## 8 Navigator

---

**Parent Class:** None

**Subclasses:** None

### Responsibilities

- Contains front end DOMs
- Transition through various tasks and tabs
- Controls the view displayed to the user
- Controls user's access

### Collaborators

- WebApp
- User
- FileSystem
- Compiler
- Shell

## 9 UserPreferences

---

**Parent Class:** None

**Subclasses:** None

### Responsibilities

- Contain data of user settings
- Manipulate data

### Collaborators

- User

## 10 FileSystem

---

**Parent Class:** None

**Subclasses:** None

### Responsibilities

- Stores files created by the user on shell
- Allows files to be imported to the user's file space
- Allows files to be exported from user's file space
- Manages the files pertaining to each user
- Displays the files pertaining to each user

### Collaborators

- Navigator
- User
- UserPreference
- Compiler

## 11 Compiler

---

**Parent Class:** None

**Subclasses:** None

### Responsibilities

- Compile given text (code)
- Communicate information to shell
- Stop compilation
- Communicate with FileSystem to upload and download files

### Collaborators

- Navigator
- Shell
- FileSystem

## 12 Shell

---

**Parent Class:** None

**Subclasses:** None

### Responsibilities

- Gets input from user
- Executes code in a live runtime environment
- Communicates with Compiler for output

### Collaborators

- Navigator

# GRACE REPL CRC Model

## Shell

- Gets input from user
- Executes code in a live runtime environment
- Communicates with Compiler for output

Navigator

## Compiler

- Compile given text (code)
- Communicate information to shell  
Stop compilation
- Communicate with FileSystem to upload and download files

Navigator  
Shell  
FileSystem

## User

- Manage user account access information
- Owns entries in the file system
- Administers shared use for owned files

FileSystem  
User-Preferences  
Navigator  
UserDAOImp

## UserPreference

- Contain data of user settings
- Manipulate data

User

## FileSystem

- Stores files created by the user on shell
- Allows files to be imported to the user's file space
- Allows files to be exported from user's file space
- Manages the files pertaining to each user
- Displays the files pertaining to each user

Navigator  
User  
User-Preference  
Compiler

## WebApp

- Authenticate user, upon filling sign in form
- Create account, upon filling register form
- Use the navigator to control application navigation

Navigator

## Navigator

- Contains front end DOMs
- Transition through various tasks and tabs
- Controls the view displayed to the user
- Controls user's access

WebApp  
User  
FileSystem  
Compiler  
Shell

## File

- Contains data
- Contains a unique file ID
- Has an list of users that can access the file
- Identifies its owner

FileDAOImp

| FileDAO   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Provide a layer of abstraction for file related database access</li> </ul> |  |

| FileDAOImp  |         |
|---|---------|
| <ul style="list-style-type: none"> <li>• Collects files in file system</li> <li>• Manages extant files in file system</li> <li>• Adds new files in file system</li> <li>• Deletes files in file system</li> </ul> | FileDAO |

| UserDAO   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Provide a layer of abstraction for user related database access</li> </ul> |  |

| UserDAOImp   |         |
|--|---------|
| <ul style="list-style-type: none"> <li>• Collects registered users</li> <li>• Facilitates user management and existence control</li> <li>• Allows for updating a user's UserPreferences</li> </ul> | UserDAO |

# System Interaction and Environment

The technologies GRACE will be using will include:

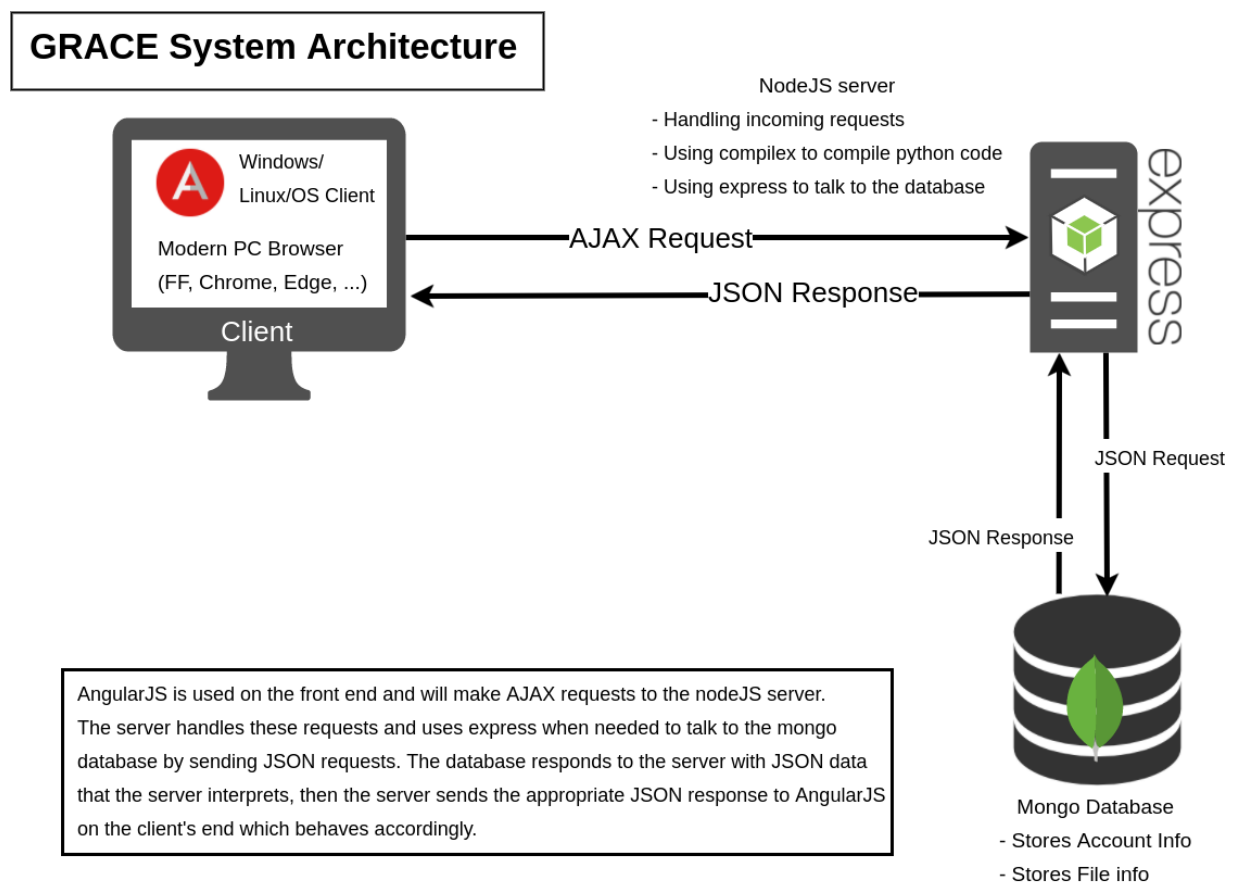
- Server: NodeJS, Express, Compilx (NodeJS library)
- Database: MongoDB
- Front-end: HTML, AngularJS, JavaScript

Most widely used PC operating systems support a modern browser such as Chrome, Firefox, or Edge.

Since these browsers support our front-end technology, and the ability to make the requests to our server, most users will be able to use our web application. Support for mobile browsers will not be present in the first release of our app but is being considered as a feature in the future.

The server will be running Linux and [MEAN](#) stack from Bitnami (version 3.4.2). NodeJS on our server will include the [Compilx](#) library (version 0.7.3).

## System Architecture





# System Decomposition

There are three main components in the GRACE REPL system architecture: client, server, and database.

The classes pertaining to the server component are File, Compiler, User, UserPreference, FileSystem, and Shell. The reason being is because the client will be making requests to the server which one or more of these classes will fulfill appropriately. More specifically, the compiler and shell in GRACE REPL will use a NodeJS library that will do the computation/interpretation on the server side. Similar to User, UserPreference, FileSystem, and File, they will be relaying on the server side receiving requests from the client while sending and receiving requests to the database.

The client component's classes are Navigator and Webapp since those classes relay the interaction between the user and the site. In addition, those interactions will essentially lead to Ajax requests to the server.

The database component consists of the classes, FileDAO, UserDAO, FileDAOImp, and UserDAOImp. Those classes deal with data access object and are in charge of storing data such as account and file information.

The system decomposition is on the system architecture diagram. It illustrates each component and its role in the higher-level architectural view.

The strategy for dealing with errors and exceptional cases will differ for each case. For the IDE and related errors such as third-party issues, the approach will be to have the users report the bug and then notify all GRACE REPL users of the existing bug. Furthermore, the GRACE team will try to resolve the issue since the Complex library is open source and notify the Complex developers.

For server connection issues, such as a user's inability to share and/or view files from other users, there are limitations for what the GRACE team can do. Connection issues occur all the time, so on our end the strategy is to prompt the user to refresh the page after a certain amount of time or wait until the connection is re-established. This strategy is the same for unexpected code compilation errors, since the server compiles/interprets the code.

Another case to consider is browser issues. Some less used web browsers may not be compatible with the site. The strategy will be to suggest the user of using a compatible browser; in the meantime, the team will try to alleviate the browser incompatibility.

Other general failures such as unexpected input to our backend will be dealt with by having multiple layers of validation that will eventually attempt to output an error message to the users.



# DESIGN MOCK-UP



\* elements are clickable

\*login

\*LOGO

language select

\*code as guest

\*register

\* elements are clickable

\*login

\*LOGO

language select

\*code as guest

Register

username

password

confirm password

email

\*create account

\* elements are clickable

\*login

\*LOGO

language select

\*code as guest

\*register

username

password

\*login

\*settings

\*LOGO

\* elements are clickable

\*logout

## Your files

↑ upload a file

📄 new file

✕ delete

🔒 make private

☒ \*file name

last modified: 01/06/2017

↓ download

✉ invite



☐ file name

last modified: 01/06/2017

↓ download

✉ invite



☐ file name

last modified: 01/06/2017

↓ download

✉ invite



☐ file name

last modified: 01/06/2017

↓ download

✉ invite



☐ file name

last modified: 01/06/2017

↓ download

✉ invite



☐ file name

last modified: 01/06/2017

↓ download

✉ invite



☐ file name

last modified: 01/06/2017

↓ download

✉ invite



\*settings

\*LOGO

\* elements are clickable

\*logout



Your username

feature



feature



feature



 change password

\*(this region is clickable)

1 replace file



new file



delete



make private

last modified: 01/06/2017

 download

 invite



last modified: 01/06/2017

 download

 invite



last modified: 01/06/2017

 download

 invite



last modified: 01/06/2017

 download

 invite



last modified: 01/06/2017

 download

 invite



last modified: 01/06/2017

 download

 invite



last modified: 01/06/2017

 download

 invite



\*settings

\*LOGO

\* elements are clickable

\*logout

↓ download

📁 save

▶ run

■ stop

✕ clear

```
1 print("Hello World")
```

Running python version 2.7.10

> Hello World!

> \_



\*back

\*LOGO

\* elements are clickable

↓ download

📁 save

▶ run

■ stop

✕ clear

```
1 print("Hello World")
```

Running python version 2.7.10

> Hello World!

> \_

\*settings

\*LOGO

\* elements are clickable

\*logout



save



run



stop



clear

\*(this region is clickable)

Running python version 2.7.10

> Hello World!

> \_

feature



feature



feature



Invite URL

<https://www.grace.com/f8s9f7>

CP

delete

new invite