**Software Implementation and Testing Document**

**For**

**Group 3**

**WeShed**

Version 1.0

**Authors**:

Steven Knudson

Eliot Shea  
Noal Gelser  
Rogelio Lopez

# Programming Languages

*Our group is using JavaScript and MySQL. JavaScript is being used for the front and back end implementations. In the front, JavaScript is being used to serve HTML via React implementations, while in the back, JavaScript plays a role in declaring listening functions and how the listening server is setup to respond and execute. The back-end server communicates with our AWS database via MySQL queries and then communicates those queries back to front end. React serves HTML through the render function.*

# Platforms, APIs, Databases, and other technologies used

*We are using an AWS RDS MySQL Database to provide persistent cloud data for our schema. The back-end talks to the AWS database and the back-end responds to front-end requests for data. The front end consists of HTML that is dynamically generated using ReactJS.*

*Packages/Libraries:*

*Front End:*

*-ReactJS*

*-ReactPlayer (component on NPM for convenient embedding of videos via url)*

*Back End:*

*-Express for creating the application*

*-body-parser for json packaging*

*-morgan for debugging*

*-cors(Cross Origin Resource Sharing) to understand front-end requests*

*-mysql to create a connection with the database*

*-crypto for encrypting passwords in the database*

*-jsonwebtoken for hashing cookie tokens*

# Execution-based Functional Testing

*Our execution-based functional testing consisted of both every member of the group testing their individual source code, as well as a team testing of the code when all members source code was integrated together.*

# Execution-based Non-Functional Testing

*Our execution-based non-functional testing consisted of both every member of the group testing any non-functional elements in their software, as well as a team testing of the non-functional requirements in a team meeting when all code was integrated together.*

# Non-Execution-based Testing

*Non-execution-based testing was done during meetings, or after class when we thought that*

*walkthroughs were needed to better understand the code. This was done many times during the first increment.*