

TEST PLAN FOR GROUP 14/TEAM TIDY

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
1.0.0	Feb/16/2023	Raheem Tihamiyu	Initial testing plan

1 Introduction

1.1 Scope

Functional requirements:

1. The system should allow users to create, update, and delete chores.
2. Users should be able to Sign up/Log in
3. Users should be able to view and manage a list of chores based on daily, weekly, monthly, and other schedules.
4. The system should allow users to create and manage their accounts so that they can access their schedules from anywhere.
5. Users should receive notifications about upcoming chores to avoid forgetting about them.

Non-Functional Requirements:

1. The system should be capable of handling high traffic, with the ability to respond to 100 users with 1000 requests per minute concurrently.
2. The system should be user-friendly, responsive, and easy to navigate.

1.2 Roles and Responsibilities

Name	Net ID	GitHub username	Role
Raheem, Tihamiyu	tihamiyao	Tosvng	Backend Developer
Syed Ali Ahsen Muhammad, Jafry	aliahses	Ahsen19	Backend Developer
Kimberly, Wills	willsk	Amikz	Frontend Developer
Joshua, Dueck	dueckj1	JoshuaDueck	Frontend Developer

Role Details:

1. Frontend Developer
 - A developer focused on creating the components that make up the UI.
2. Backend Developer
 - A developer focused on the database controllers and links backend to frontend.

2 Test Methodology

2.1 Test Levels

Core feature 1: Create, update, and delete chores

Unit Tests:

- ii. Make sure new chores are created with the appropriate fields
- iii. Make sure only users with valid accounts can create a new chore.
- iv. Make sure creating a chore that already exists returns 409 and should prompt the user to use the edit functionality.
- v. Make sure to respond with 500 if there is an issue with the server.
- vi. Make sure users are able to view a newly created chore.
- vii. Make sure users are able to retrieve all the chores they have created.
- viii. Retrieving a single chore should be possible.
- ix. Users should be able to delete chores.

- x. Users should be able to edit existing chores.
- xi. Make sure only registered users should be able to make CRUD requests
- xii. Make sure GET request routes to the appropriate controller.
- xiii. Make sure POST request routes to the appropriate controller.
- xiv. Make sure PUT request routes to the appropriate controller.
- xv. Make sure DELETE request routes to the appropriate controller.

Core feature 2: Sign up/Log in

Unit Tests:

- i. Make sure creating a user object with email and password should be possible.
- ii. Make sure user object has an email and password field.
- iii. Make sure users can register with an email and a password.
- iv. Make sure users should not be able to register with the same email twice.
- v. Make sure users can login with the correct email and password
- vi. Make sure users with an incorrect email cannot log in.
- vii. Make sure users cannot log in with an incorrect password.
- viii. Make sure successful log in requests are responded to with a 201 status and a JSON web token.
- ix. Successful register requests should be responded to with a 200 status and a JSON web token.
- x. Make sure to inform the user when there is an issue with the server and return a 500 status.
- xi. Make sure POST requests route to the appropriate controllers

2.2 Test Completeness

Criteria for test completeness:

- Automated testing should be executed
- All the bugs and failed tests should be checked and fixed regularly
- 100% code coverage
- Bugs should be fixed and reported

3 Resource & Environment Needs

3.1 Testing Tools

The following tools are being used in this project for testing:

- Automation: GitHub Actions
- Backend test: Mocha, Chai, Sinon
- Other: Postman

3.2 Test Environment

- Windows 11 and above
- Node 18.x

4 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
JSON	JavaScript Object Notation
CRUD	Create, Read, Update, Delete