NYU TANDON SCHOOL OF ENGINEERING

Computer Science and Engineering

Chores App

System Requirements Specification (SRS)

Version 1.0

Document Number: SRS-001

Project Team Number: A08

Project Team Members: Helen Xu (hjx201), Kevin Grajeda (kag622), Alwyn Zhang (az1436)

REVIEW AND APPROVALS

<Team Members>	Function (Author, Reviewer, Approval)	Date	Signature
Alwyn Zhang	Author	10/07/2020	Alwyn Zhang
Kevin Grajeda	Author	10/07/2020	Kevin Grajeda
Helen Xu	Author	10/07/2020	Helen Xu

REVISION LEVEL

Date	Revision Number	Purpose
10/08/2020	Version 1.0	Initial Release

TABLE OF CONTENTS

1. Document Purpose

1.1 Purpose

The purpose of this software requirements specification document is to be a guiding tool for the developers to use to create the Chores App. It clarifies the system's specifications and function/non-functional and business requirements. This document is intended for the development team, testing team, and the clients of the system.

2. Introduction

2.1 Scope

The system will give users the ability to manage their shared tasks and responsibilities with their roommates/housemates. Since people often put off responsibilities or forgot about them, this application aims to help users stay on top of their tasks and to communicate with roommates/housemates effectively. Users can create tasks that can be seen by other users within the application and they can be modified by the user based on their status. The system is not intended to help the user complete specific tasks/chores, but to be a management tool resource for users.

2.2 Identification

Identification: SRS-001, Version 1.0

2.3 Bounds

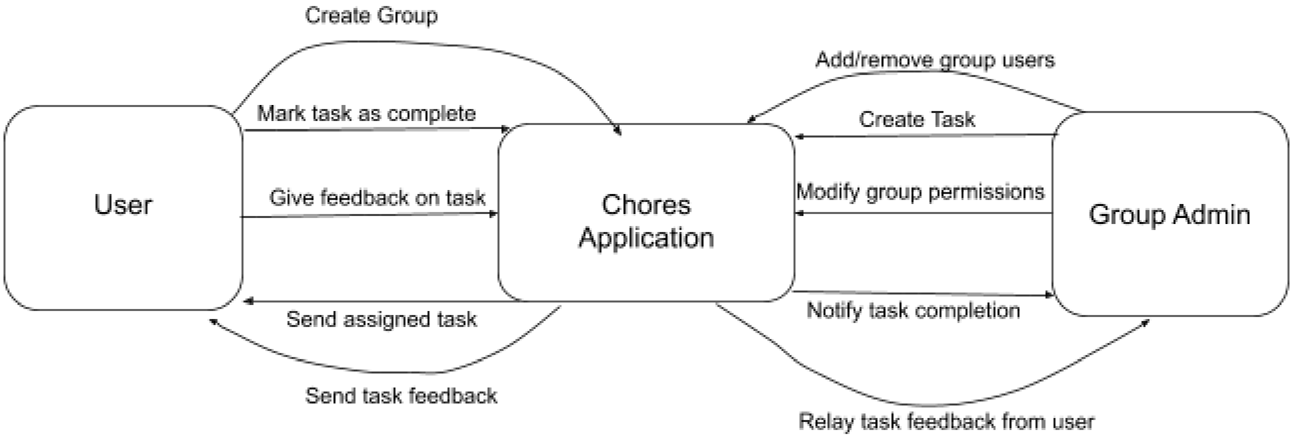
The system will require an internet connection in order for users to access it. It will also require that users create accounts using their email address and to set up a username and a password.

2.4 Objectives

The system will be developed using a combination of the Agile development model and a plan-driven model. The project's development started with the project proposal. It will continue with the development of the SRS, SPMP, and a presentation of the project itself. As for the system's development, the main priority is to create the task creation and management system. The next priorities are implementing group interaction and user account creation. The UI design has a low priority. The life cycle of the system is incremental, with the highest priority parts of the system being delivered first. The initial milestone dates are:

Deliverable	Delivery Date
Project Proposal	September 22, 2020
Software Specifications Requirement(SRS) - Domain Definition	October 8, 2020
SRS - Project Requirements	October 20, 2020
Software Project Management Plan (SPMP)	November 2, 2020
SRS - Project Analysis	December 1, 2020
Presentation	Final Two Weeks of Semester

2.5 Context Diagram



Users become group administrators when they create a group or are appointed by an administrator of an existing group. Depending on the group permissions set by the group admin, users with the group may anonymously submit feedback on tasks completed by other users, which can be sent to the admin or directly to the user responsible for the task.

2.6 Additional Descriptive Items

Product Functions:

-Allow users to create accounts

-Group creation and moderation

-Task/chore creation

-Task feedback

-Task history

-Homepage

User Characteristics:

-Regular web and mobile application experience

Constraints:

-Compatibility between web and mobile interfaces

Assumptions and Dependencies:

-Modern web browser

-Modern cell phone with internet connection

3. Glossary

None at this time. May be added in a future release.

4. Reference Documents

Project Proposal, Xu, Grajeda, Zhang, September 2020

5. Business Requirements

5.1 Technology

The application supports the business goals and objectives by appealing to households who want to organize and assign chores more efficiently and less directly.

5.2 Economics

The demand for the services provided by the application comes from families and roommates who seek to streamline managing their chores. Assigning chores is often found awkward and can feel confrontational, and the application helps users avoid such situations.

5.3 Regulatory and Legal

The regulatory business requirement we have is maintaining the privacy of users. We want to ensure that the important user information such as email address and password are protected. The application should also be secure and free from any possible exploits.

5.4 Market Considerations

Currently there exist several task management softwares made for roommates/households. While they provide intuitive interfaces to organize and schedule tasks, our app also facilitates communication by providing a system for easier communication amongst groups and the ability for users to provide feedback on tasks completed by other group members.

5.5 Risks and Alternatives

No.	Title	Est. Likelihood of Occurrence	Est. Impact	Est. Cost of Mngmt.	Priority No.	Retirement Plan	Responsible Person	Target Completion Date
1.	User falsely reports task as complete	5	7	6	1	Completed tasks show up in a history log	Kevin Grajeda	Rolling basis
2.	User falsely or accidentally deletes task	3	7	6	2	Deleted tasks show up in a history log	Alwyn Zhang	Rolling basis
3.	User creates same chore twice	4	5	3	3	Check if a similar tasks exists	Helen Xu	Rolling basis

5.6 Human Resources and Training

The engineering team will have sufficient training to be qualified to develop all the components of the system. The entire team must be familiar with all of the project’s documentation.

6. User Requirements (Descriptive Functional and Non-functional requirements)

To be added in a future release.

This section should contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test the system that meets those requirements. Every requirement should be externally perceivable by users, operator, or external systems.

The following principles should be applied:

a. specific requirements should be stated in conformance will all the characteristics described in earlier documents (for example the proposal) or earlier sections of the document

b. specific requirements should be cross-referenced (traceable) to earlier documents that they relate

c. all requirements should be uniquely identifiable

d. careful attention should be given to organizing the requirements for readability and understandability

6.1 Functional Descriptive Detailed Requirements

To be added in a future release.

6.2 Non-Functional Descriptive Detailed Requirements

To be added in a future release.

7. System Architecture

To be added in a future release.

This section should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.

8. Detailed System requirements – Use Cases

(Chapter 5 – Project Requirements Deliverable)

8.1 Requirement Use Cases

To be added in a future release.

This subsection contains a complete set of Use Cases, ranging from high level views to low (decomposed) levels. Every use case contained in a Use Case diagram must have a respective Use Case description.

8.1.1 Use Case Diagrams

To be added in a future release.

Below is a sample use Case diagram - in your diagrams include the boundary rectangles

8.1.2 Use Case Descriptions

To be added in a future release.

Use Case Name		
Description		
Pre-Conditions		
Flows	**Basic or Normal Flows**	1. 2. 3.
	Alternative Flows	1. 2. 3.
Post Conditions		
Special Requirements		
Extension Points		

9. System Model (UML)

To be added in a future release.

Graphical system model

9.1 Static - Class Diagrams

Candidate classes (list)

Class Diagram (without interaction)

Class Interaction Diagrams (class interaction, relationship, and cardinality)

9.2 Dynamic - Behavioral Models

Event diagrams (One per each Use Case/Functional Requirement)

Sequence Diagrams (One per Use Case/Functional Requirement – must cover all classes)

Collaboration Diagrams (One for the system flow)

10. Evolution of the SRS

The SRS will be updated when:

1. Requirements are changed, added or removed
2. If inaccurate information or shortcomings are found
3. When deliverables are completed

Any changes to the SRS will be reported; the changes will be checked and approved by everyone in the group and the version number updated when there is a change.

11. Rationale

None currently. May be added in a future release.

12. Notes

None currently. May be added in a future release.

13. APPENDICES

13.1 System Test Plan Requirements

In this section describe the SQA testing process, product test and acceptance testing

The SQA testing process will be used to evaluate the system to ensure that the requirements specified in this document are being met. It will also evaluate the system's security and reliability.

To test each requirement, the system will be run through a test scenario similar to a use case involving that requirement. The results of the test will be compared to a standard set for that requirement before the test.

In order for the software to be reliable, it needs to be able to have a 99% uptime with an average to high amount of traffic. This will be tested by running the application with simulated traffic.

It must also be tested for security, as the software should not be easily hacked. The application will be tested for any vulnerabilities through ethical hacking.

Test plans with scenario testing and required simulators

13.2 Qualification Provisions

This document will be reviewed for correctness, completeness, and consistency. It will be reviewed both individually and as a group to ensure that it is unambiguous, stable, modifiable, verifiable, and traceable.

If a defect is discovered during a review, it will be brought to the group for verification and a group member will be assigned to fix the defect. Once it has been fixed, the other group members will review the document to affirm the defect has been resolved. If the group determines that the defect has not been fixed, the group will discuss how to best resolve the issue and modify the document accordingly upon coming to an agreement.

13.3 Requirements Traceability

To be completed in a future release

This section describes how each requirement is traceable from its original source to instances in each succeeding artifact. It must facilitate referencing of each requirement in future development or enhancement documentation. Two types of traceability are forward (to all artifacts spawned by this document) and backward (to previous stages of development).

13.4 Schedule Tracking

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS Domain	Alwyn Zhang	3 hours	2 hours	1 hour
SRS Domain	Kevin Grajeda	4 hours	2 hours	2 hours
SRS Domain	Helen Xu	3 hours	1 hour	2 hours
	Entire Team	10 Hours	5 hours	5 hours

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS Requirements				

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS Analysis				

Cumulative

Who (individual or Team)	Estimated	Actual	Difference
Alwyn Zhang	3 hours	2 hours	1 hour
Kevin Grajeda	4 Hours	2 Hours	2 hours
Helen Xu	3 hours	1 hours	2 hours
Entire Team	10 Hours	5 hours	5 hours

13.5 Defect Tracking

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
Initial SRS	Alwyn Zhang	5	6	1
Initial SRS	Kevin Grajeda	12	8	4
Initial SRS	Helen Xu	7	5	2
	Entire Team	24	17	7

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
Final SRS				
	Entire Team			

Cumulative

Who (individual or Team)	Estimated	Actual	Difference
Alwyn Zhang	5	6	1
Kevin Grajeda	12	8	4
Helen Xu	7	5	2
Entire Team	24	17	7

14. index

To be added in a future release.