

2019 Fall Term

CS 561 Project

Remote Control Automatic Door System

Beisi Yan
Mengchuan Niu
Mikhail Tokarev
Yuxuan Che
Zixuan Feng
Zeyuan Chen



Oregon State University
College of Engineering

Oregon State University
School of Electrical Engineering and Computer Science
CS 561 Term Project
10-XX-2019

Contents

1	Introduction	1
2	Development Process	2
2.1	UML Diagrams	2
2.2	User Stories	3
2.2.1	Home page	3
2.2.2	Sign-in	3
2.2.3	Sign-up	3
2.2.4	Rent a storage	3
2.2.5	Return a storage	4
2.2.6	My storage page	4
2.2.7	Page for administrator	4
2.2.8	User account Page	4
2.2.9	Storage mapping page	4
2.2.10	Sending the password	5
2.3	First spring user stories	5
2.3.1	Home page	5
2.3.2	Sign-in	5
2.3.3	Sign-up	5
2.3.4	Rent a storage	6
2.3.5	Sending the password	6
2.4	Second spring user stories	6
2.5	Third spring user stories	6
2.6	Development Languages and Tools	6
3	Developing Results in Progress	8
3.1	First spring	8
3.2	Second spring	8
3.3	Third spring	8
4	Conclusion and Limitation	9

5	Appendix	10
5.1	Evidence of the daily Scrum	10
5.2	Evidence that you held a Sprint Review with users	10
5.3	Evidence that your team held a Sprint Retrospective	10
	Bibliography	11

List of Figures

2.1	Software flow chart	2
2.2	Abstract first sprint flow chart.	6
2.3	Detailed first sprint flow chart.	7

List of Tables

1. Introduction

Renting storage and warehouse are needed in daily lives among some families, such as military families, college students, and people who are planning long term travelling. Within the last decades, self renting storage is getting popular [2]. In 1999, automated self-storage has been introduced by Kanoh and Itoh [1], which describe an automated self-storage system can work 24 hours 7 days. As they described, the user can go to the storage company, by using the machine to rent the storage and acquire the keys from that machine without storage staff involved. However, it still not convenient for users, such that user still need to go to the storage physically to check the availability and rent the storage. In the modern business market, companies can rely on the internet to seek more portable and convenient ways to manage the business process, as well as provide convenience for users. In this project, we focusing on developing a website of the remote control automatic door system for stakeholders who are seeking systemically and organized way to manage storage and warehouses as well as providing a convenient method for their users.

Our website including the database system management method proposed to a self-management of the door control system. By log into our website, the user can find the local storage based on storage map regard to the user locations. After selecting the storage as needed, the user would receive a randomized password which is for open the storage during the renting period. User in our website does not need to go to the storage office by online booking and checking in instead, the user may access and rent the storage without time and location limitations. In addition, as stakeholders, company can manage the storage remotely, which saves a lot expenses and labors.

2. Development Process

2.1 UML Diagrams

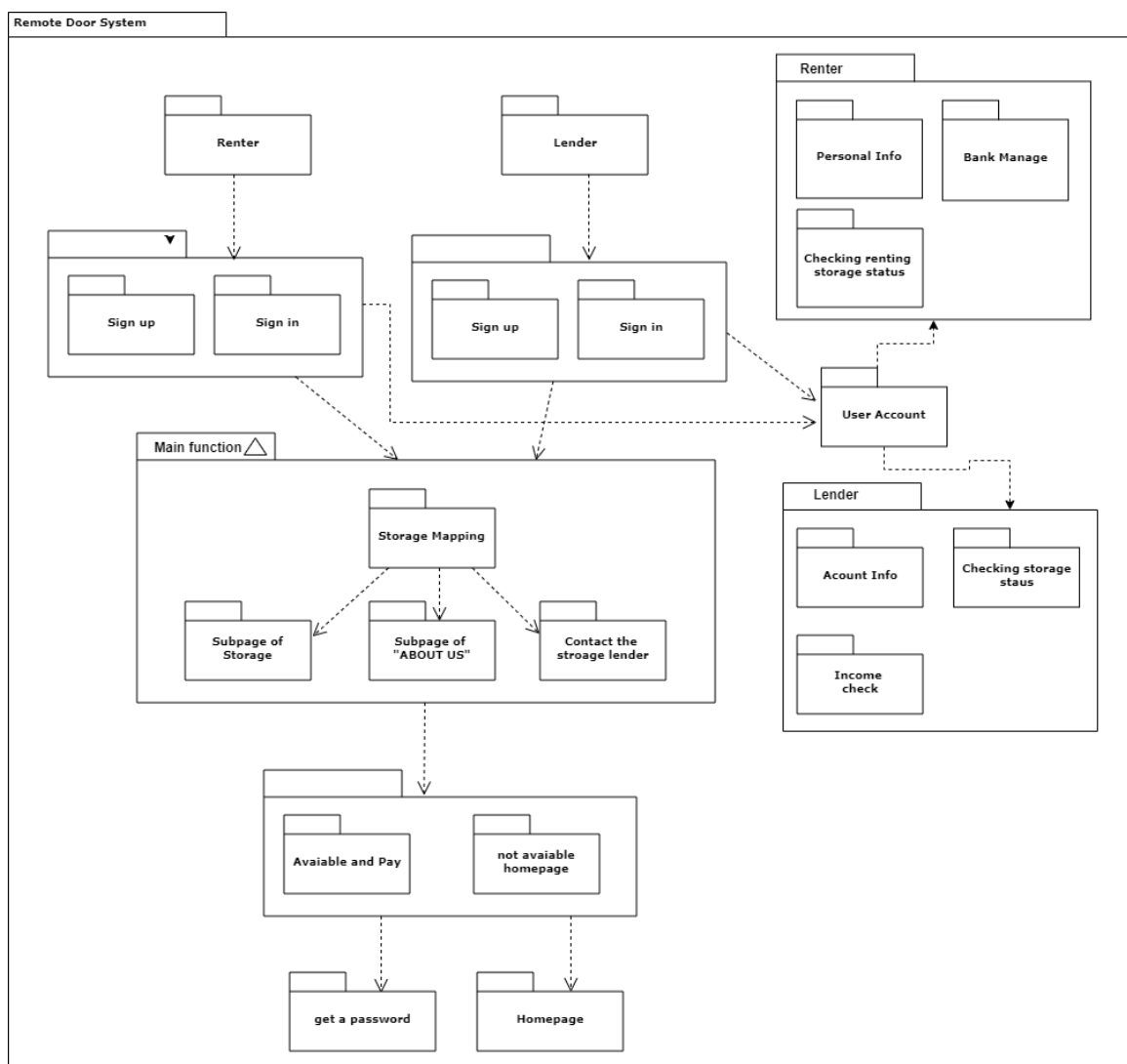


Figure 2.1: Software flow chart

2.2 User Stories

2.2.1 Home page

- Website Homepage (Briefly introduce the website in 1 or 2 sentence, and UI designs)
- About us page (Company missions)
- Log-in and Sign-up interactive button
- contact with the website owner button

Effort estimate: TBD

2.2.2 Sign-in

- Login page (including UI design)
- Check the username and password
- Chose the characteristics (renter or lender, this will lead to two different main-function pages)
- Create test verifies login works.
- Back homepage button
- Sign-up button

Effort estimate: TBD

2.2.3 Sign-up

- Sign up page (Including UI design)
- Required information list
- Back homepage button
- Database organization (database design and maintenance)

Effort estimate: TBD

2.2.4 Rent a storage

*renter

- "main-function" page (rent the storage that the renter chose)
- database management of request interaction (Create code to write users information into the database then give a UID)
- Unit test

Effort estimate: TBD

2.2.5 Return a storage

- return page (including UI)
- database management of request interaction (Database update)

Effort estimate: TBD

2.2.6 My storage page

- check how many days left
- extend storage page
- database management of request interaction (Database update)

Effort estimate: TBD

2.2.7 Page for administrator

*lender

- check the storage status (how many storage is available, how many storage is empty, and how many storage is reserved by empty)
- database management of request interaction
- check each individual profit

Effort estimate: TBD

2.2.8 User account Page

- Account information page (personal information, bank information, upload the image)
- personal information page
- bank information page
- database management of request interaction

Effort estimate: TBD

2.2.9 Storage mapping page

- Mapping page list all the storage near the user (similar with YELP)
- sub-page of the storage that the renter is interested
- database management of request interaction

Effort estimate: TBD

2.2.10 Sending the password

- Looking for a service that once the renter click the button, the password will send through the text

Effort estimate: TBD

2.3 First spring user stories

2.3.1 Home page

- Website Homepage (Briefly introduce the website in 1 or 2 sentence, and UI designs)
- About us page (Company missions)
- Log-in and Sign-up interactive button
- contact with the website owner button

Requirements: HTML, CSS, JQuery, finished up the basic functionality of the homepage.

2.3.2 Sign-in

- Login page (including UI design)
- Check the username and password
- Chose the characteristics (renter or lender, this will lead to two different main-function pages)
- Create test verifies login works.
- Back homepage button
- Sign-up button

Effort estimate: TBD Requirements: HTML, CSS, JQuery, finished up the basic functionality of the sign-in page. Ensuring the basic functionality of the database.

2.3.3 Sign-up

- Sign up page (Including UI design)
- Required information list
- Back homepage button
- Database organization (database design and maintenance)

Effort estimate: TBD Requirements: HTML, CSS, JQuery, finished up the basic functionality of the sign-up page. Ensuring the basic functionality of the database.

2.3.4 Rent a storage

*renter

- "main-function" page (rent the storage that the renter chose)
- database management of request interaction (Create code to write users information into the database then give a UID)
- Unit test

Effort estimate: TBD Requirements: HTML, CSS, JQuery, finished up the basic functionality of the renting page. Ensuring the basic functionality of the database.

2.3.5 Sending the password

- Looking for a service that once the renter click the button, the password will send through the text

Effort estimate: TBD Requirements: HTML, CSS, finished up the basic functionality of the password page. Ensuring the basic functionality of the database. Ensure user can receive the password

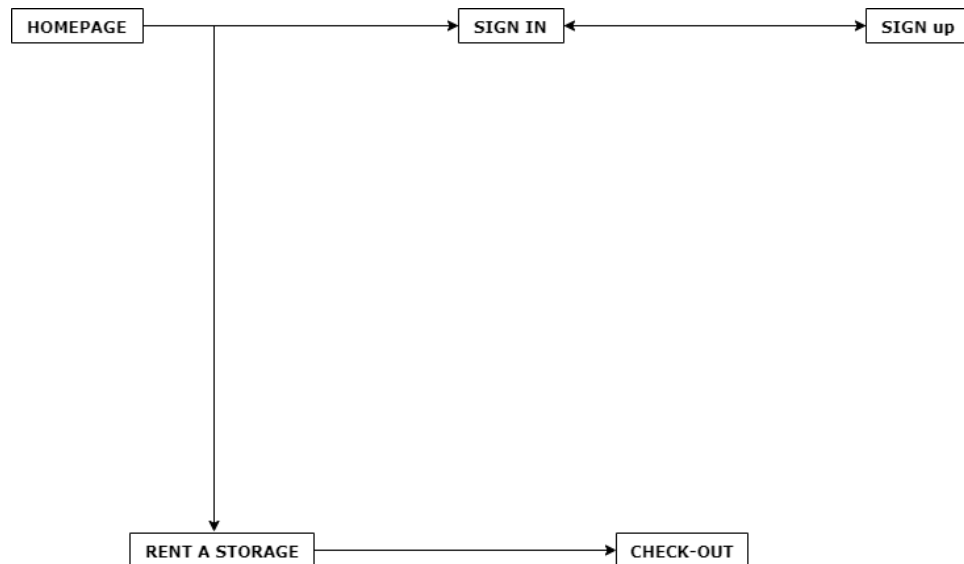


Figure 2.2: Abstract first sprint flow chart.

2.4 Second spring user stories

2.5 Third spring user stories

2.6 Development Languages and Tools

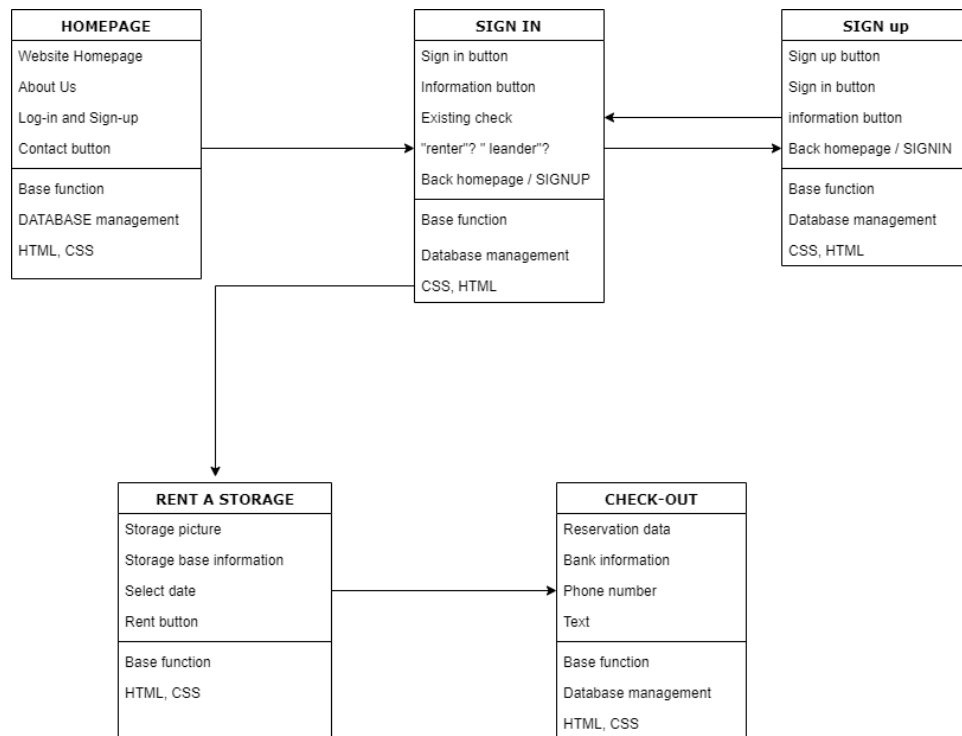


Figure 2.3: Detailed first sprint flow chart.

3. Developing Results in Progress

3.1 First spring

3.2 Second spring

3.3 Third spring

4. Conclusion and Limitation

5. Appendix

5.1 Evidence of the daily Scrum

Github https://github.com/zixuanfeng/CS561/tree/master/meeting_log

5.2 Evidence that you held a Sprint Review with users

5.3 Evidence that your team held a Sprint Retrospective

Bibliography

- [1] Charles H McCarty and Bonita S Linder. Automated storage system, August 31 1999. US Patent 5,946,660.
- [2] Scott Meyers. Why is self storage so popular?, 2018.