SCMS Worksite E-Form System

Team 22

Kenneth, Yuan Lin, Yu Chyi, Kin Leung, Johnathan

Cloud Plus

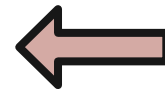Account Login

Username

Password

✓ Remember me   Forgot Password?

Login
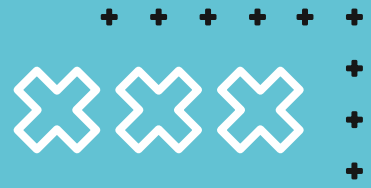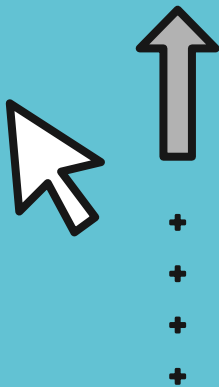
Agenda

1. Introduction
2. Gantt Chart
3. System Architecture
4. Achievements & Product Features
5. Live Demonstration
6. Automated Test
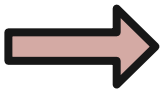7. Challenges Faced
8. Future Work
9. QnA

01
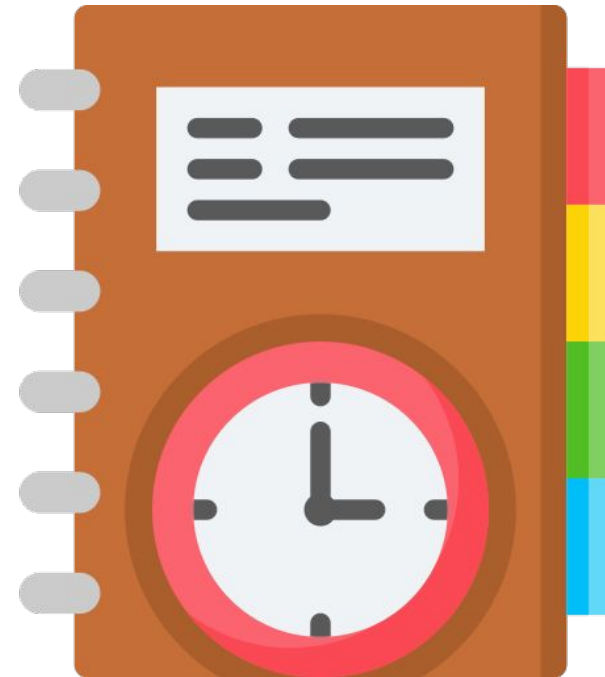
Introduction

Introduction

Background

- Many forms to fill up at the worksite
- Implement an E-Form system to replace forms for worker to fill in

Motivation

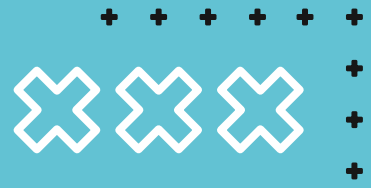- Replace paper forms to streamline the workflow in construction worksites
- Increased operational efficiency
- Reduce storage costs
- Reduce retrieval time

Goal

- Better organization of forms for project managers and workers
- Forms can be categorized and have a centralized location.
- Provide convenience to both project managers and workers.
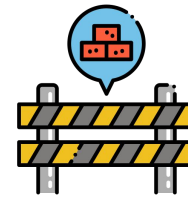
02

Gantt Chart
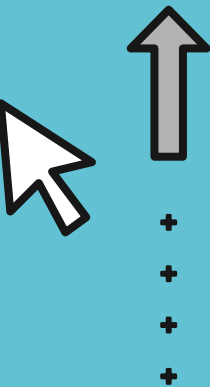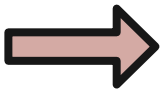
03

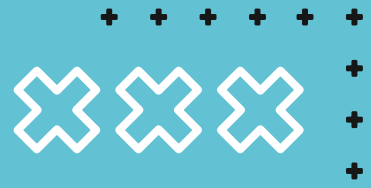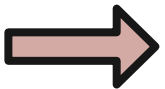System Architecture
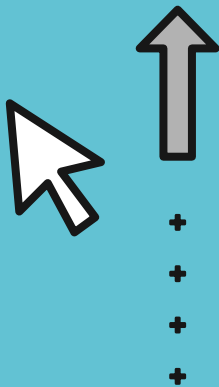
System Architecture
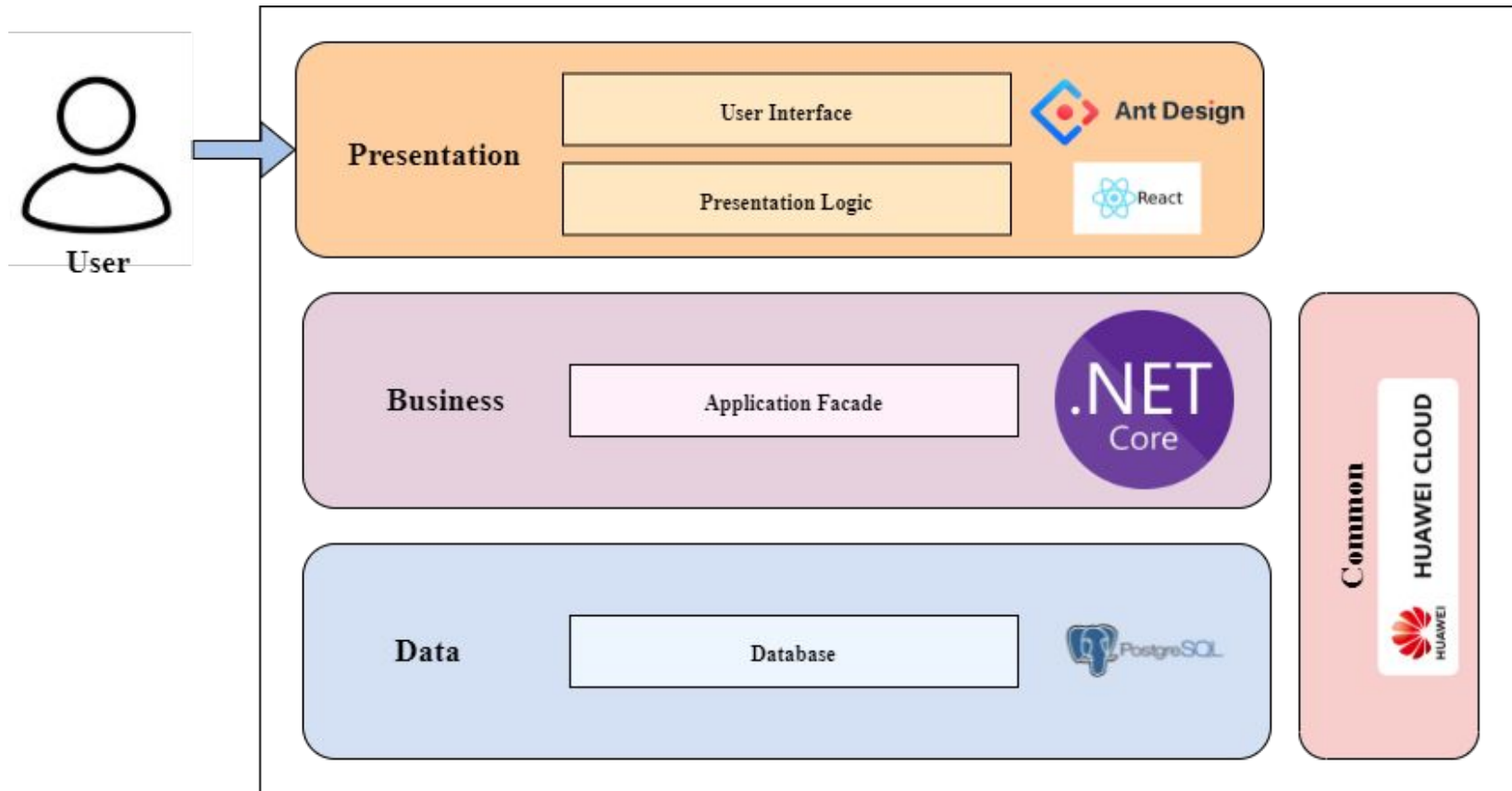
High Level Software Architecture

Client-Server Architecture Pattern

Users — Website — Database

Worker 1, 2, 3, …
Manager 1, 2, 3, …

Application Server

Database Server

04
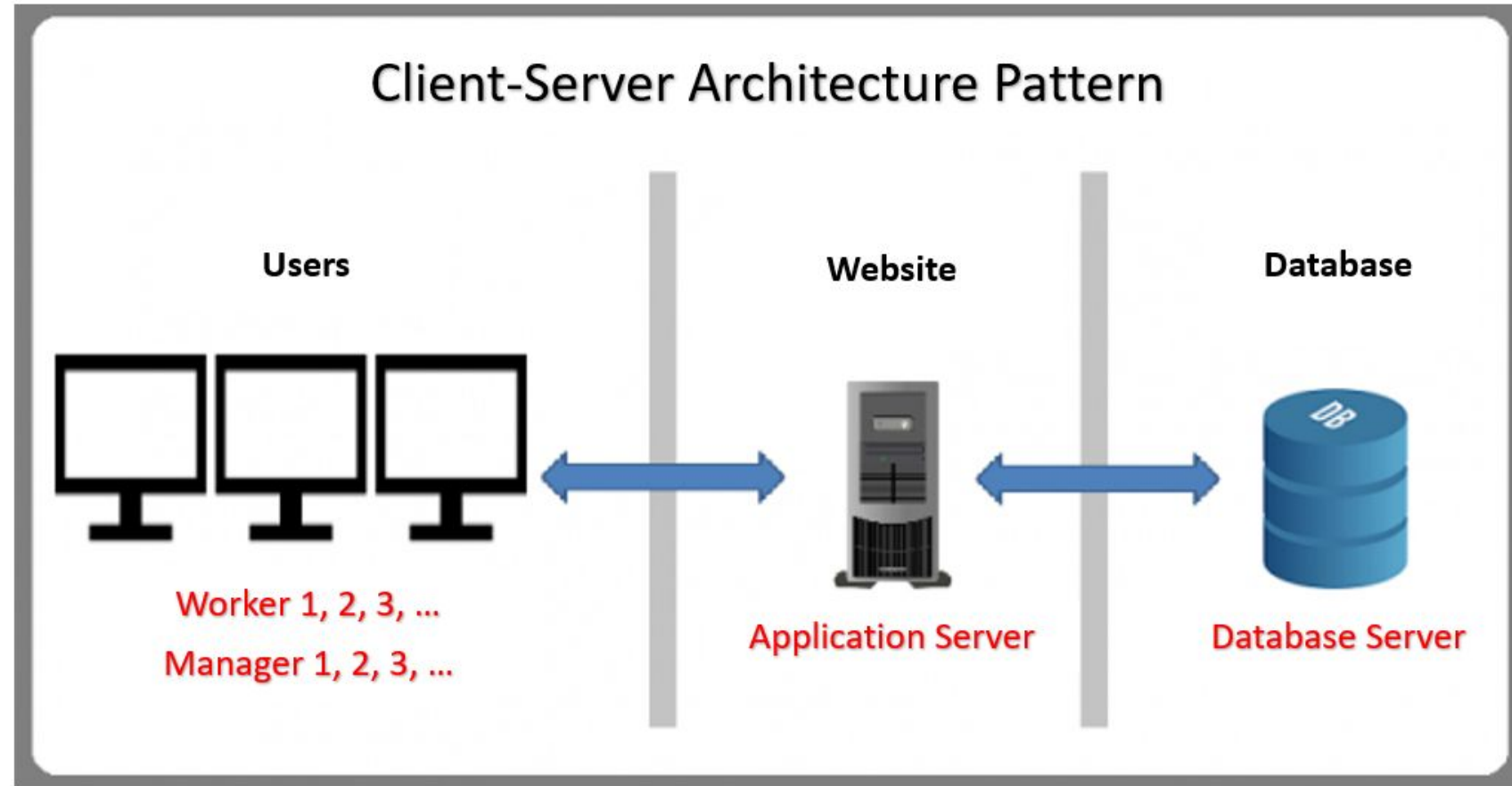
Features & Achievements

Manager

1. Project Management

Project Management

Project ID	Project Name	Description	Actions
2	Create Project	Create Project 1	Edit View Delete
1	Create Project	Create Project 1	Edit View Delete

2. Forms Management

Create Project Search [Enter Form Name]   Sort By [Descending ∨]   Create Form

#2 - form 2
form 2   Manage

#1 - new form
new form   Manage

Manager

3. Form Customisation

Form Details

Form Title:

Form Description:

Form Expiry Date: Select date 📅

Load From Template

Form Template: Choose template ⌄   Load

Customise Form

Question 1: Is this a question?

Input Field

Required? 🗑 ︿ ⌄

Question 2: Another question but now it's required?

Input Field

Required? 🗑 ︿ ⌄

Add Element: Input Field ⌄   +

Create Form   Cancel

Elements Implemented:

- Textbox
- Text Area
- Numeric Field
- Checkbox
- Radio Button
- Date Picker

Manager

4. Pending Review

Pending Review

Submission #3
Details

Form Name: form 2
Project Title: Create Project
Submitted Date: 2022-03-17
Submitted By: 0

| Approve | Reject |

Pending Review

Submission #3
Details

Form Name: form 2
Project Title: Create Project
Submitted Date: 2022-03-17
Submitted By: 0

Enter a reason for rejection:

| Cancel | Confirm Rejection |

Manager

5. Form Listing

Form Listing

Submission #2
Details

Form Name: form 2
Project Title: Create Project
Submitted Date: 2022-03-17
Submitted By: 0

⊘ Approved

Submission #1
Details

Form Name: new form
Project Title: Create Project
Submitted Date: 2022-03-17
Submitted By: 0

⊗ Rejected

Reason: invalid

Manager

6. Archived

Worker

1. Actions Required

Form Name	Description	Expiry Date	Actions
☐ 1 form 2	form 2	2022-03-20	Fill Form
☐ 2 new form	new form	2022-03-19	Fill Form

Form Name: _Please enter_   Description: _Please enter_   Expiry Date: _Please enter_

⌂ / Action Required

Worker

2. Form Management

Submission #1

Form Name: new form
Project Title: Create Project
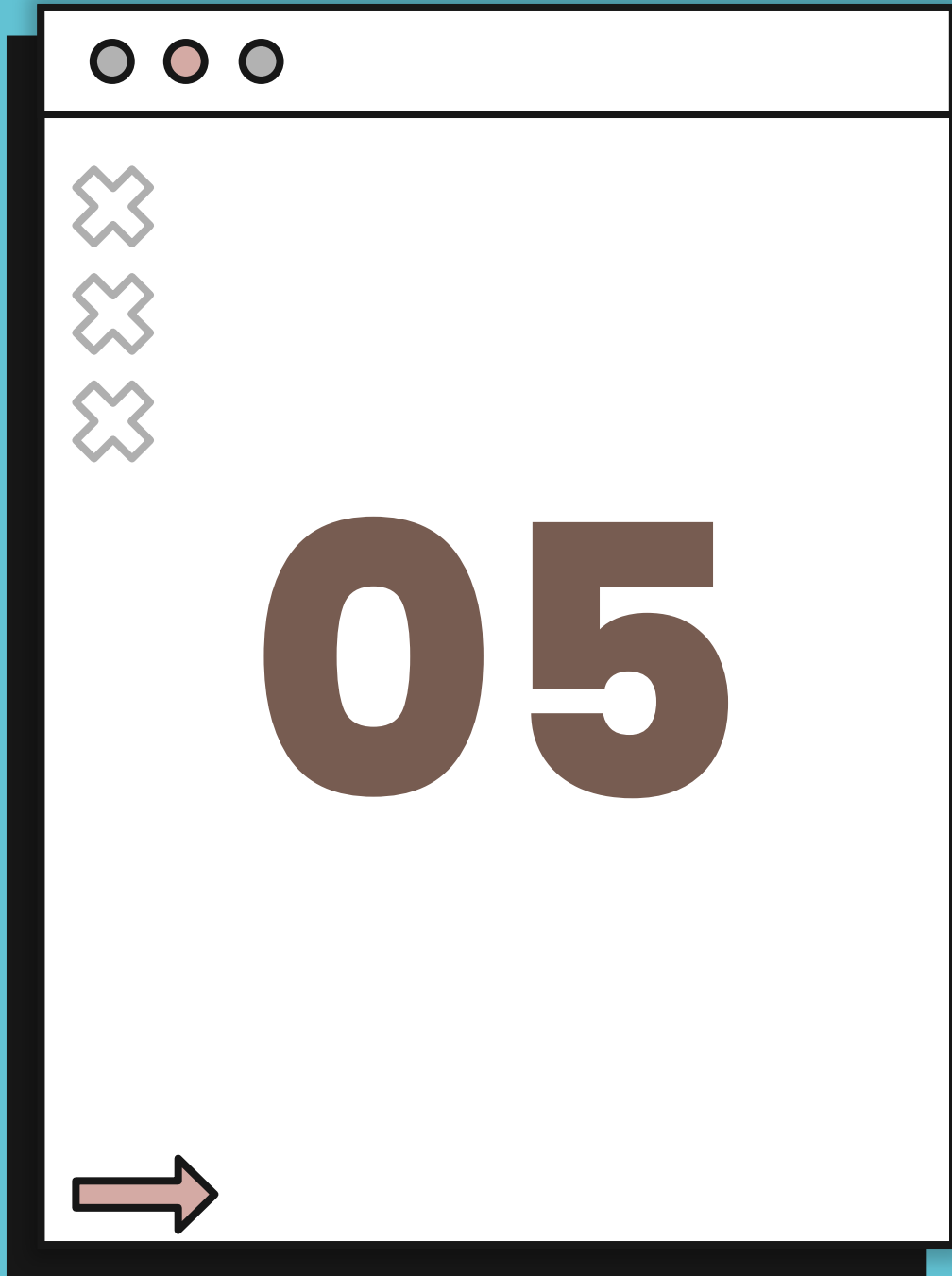**Submitted Date:** 2022-03-17
Status: Rejected

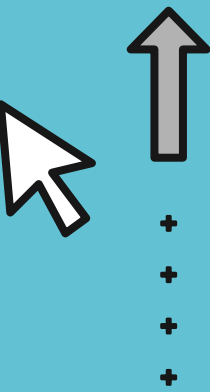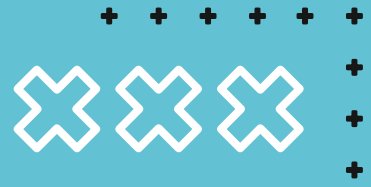**Rejection Reason:** invalid

Form Management

	Submission ID:	Please enter	Form Name:	Please enter

	Submission ID	Form Name	Status	Actions
1	3	form 2	Pending	View Details
2	2	form 2	Approved	View Details
3	1	new form	Rejected	View Details
4	0	Test	Approved	View Details

05
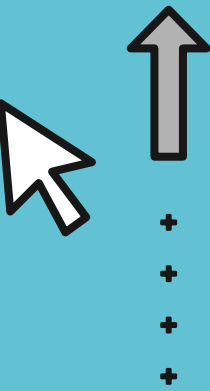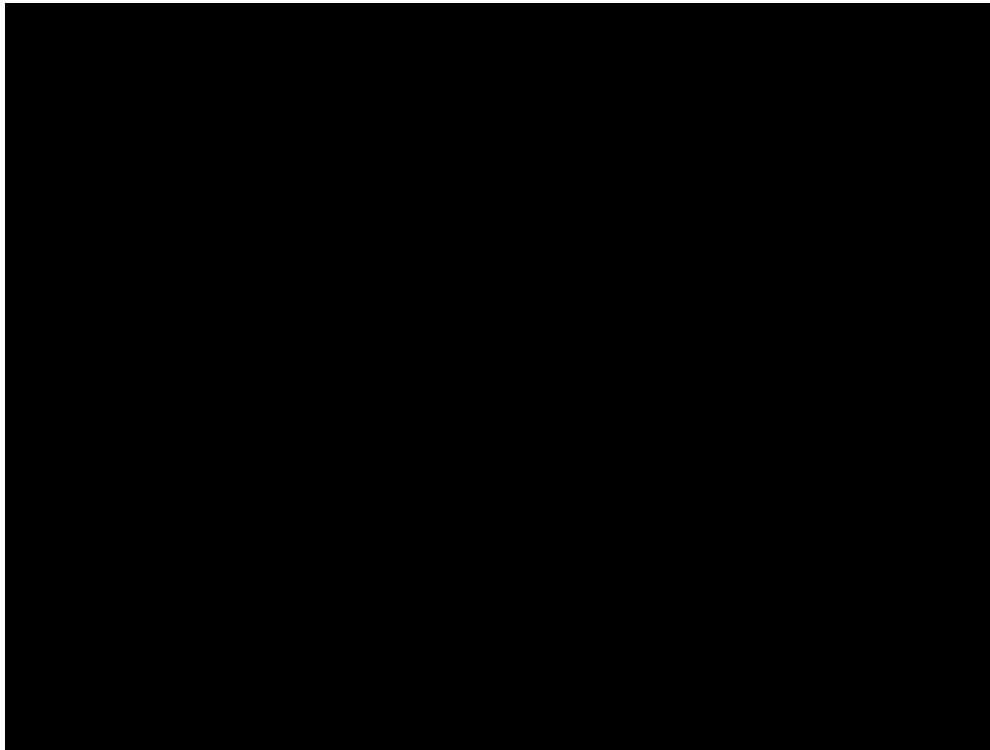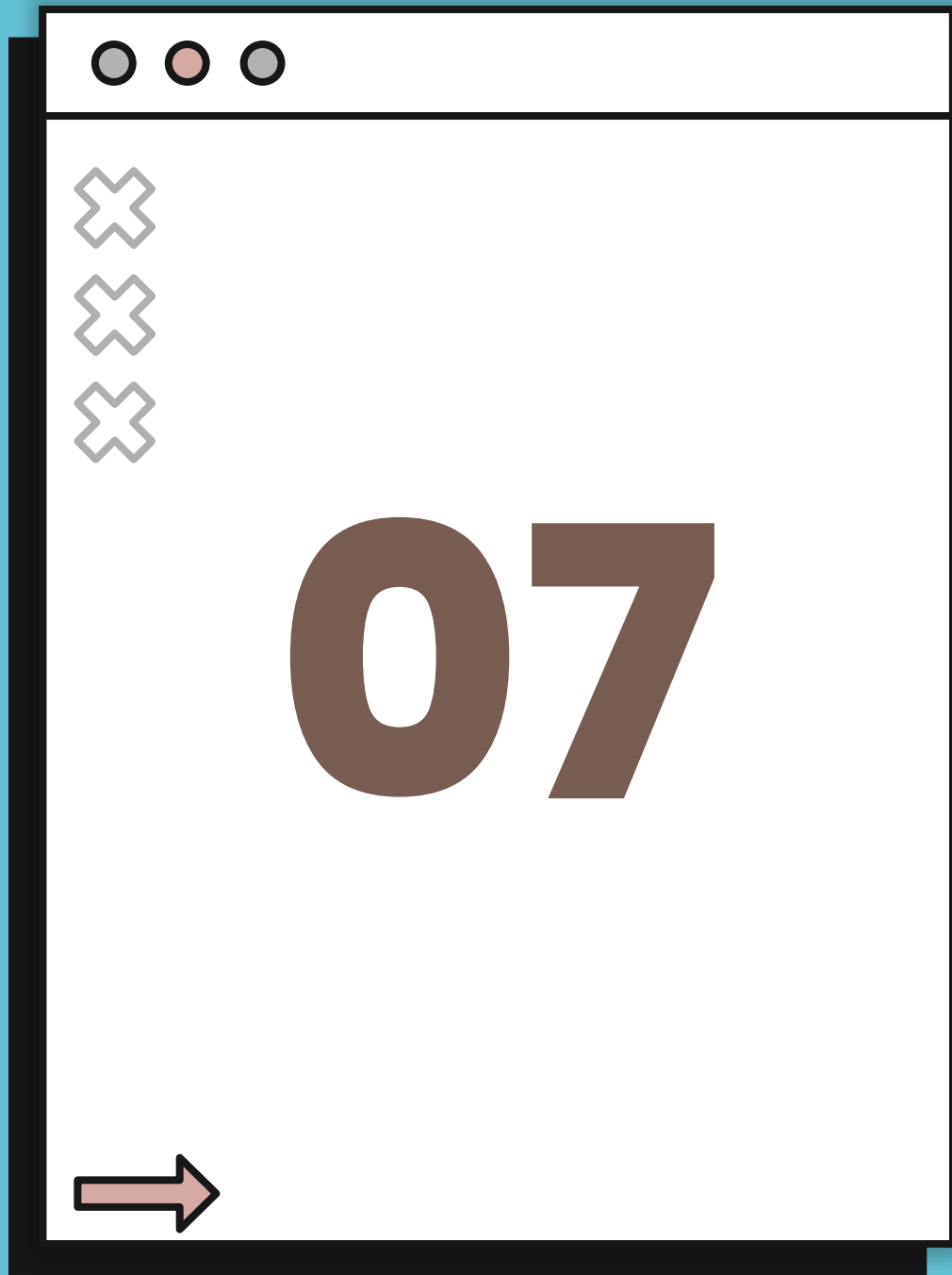
Live Demonstration

06
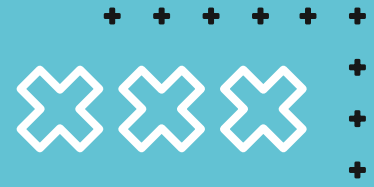
Automated Testing

Automated Testing

07

Challenges Faced

Challenges Faced

Technical Challenge #1

Linking backend of .NET with the frontend

- GET, POST, PUT and DELETE requests could be made
- Unable to pass additional parameters to backend

Solution:

- Separate internal lightweight app to make interactions and test endpoint
- External applications (e.g. Postman) for debug
- Migrate working codes to main project

GET
PUT
POST
DELETE

POSTMAN

Challenges Faced

Technical Challenge #2

Each question in form contains a different type of form element

- E.g. Input field and checkbox field are two different types of form element
- Challenging to pass updated state of components to form builder
  - Especially difficult for questions with nested states (e.g. Checkboxes & Radio Buttons)

Solution:

- Helper functions to update the state of each type of components

```javascript
const updateQuestion = (position, question) => {
    let newFormElements = [...formElements];
    newFormElements[position].question = question;
    setFormElements(newFormElements)
}

const updateCheckboxOptions = (position, checkboxOptions) => {
    let newFormElements = [...formElements];
    newFormElements[position].checkboxOptions = checkboxOptions;
    setFormElements(newFormElements)
}

const updateRadioOptions = (position, radioOptions) => {
    let newFormElements = [...formElements];
    newFormElements[position].radioOptions = radioOptions;
    setFormElements(newFormElements)
}

const updatePosition = (currentPosition, delta) => {
    let newFormElements = [...formElements];
```

```jsx
eturn (
    <BuilderCheckboxElement
    key={e.key}
    position={e.position}
    question={e.question}
    required={e.required}
    options={e.checkboxOptions}
    questionListSize={formElements.length}
    onChangeRequired={(position, value) => {
        updateRequired(position, value)
    }}
    onChangeQuestion={(position, value) => {
        updateQuestion(position, value)
    }}
    onChangeCheckboxOptions={(position, value) => {
        updateCheckboxOptions(position, value)
    }}
    onTriggerPositionChange={(position, changeValue) => {
        updatePosition(position, changeValue)
    }}
    onTriggerDelete={(position) => {
        deletePosition(position)
    }}
    />
```
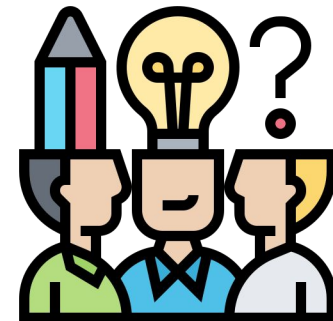
Challenges Faced

Deep Learning Curve

- The team have little experiences on technical requirements
  - React JS [Ant Design Pro]
  - .Net Core
  - PostgreSQL
- Takes time to learn and apply what the team learn

Overcome:

- Analyse different groupmates strengths and weaknesses
- Split job scope accordingly to ensure maximum efficiency
- Guide one another to ensure everyone learns together

Challenges Faced

Project Integration

- Standardisation of project design between teams is time-consuming
- Projects managed by every teams all have different progress
    - Bug-fixes, revisions, quality control
- Uncertainties in project standardisation and integration

Overcome:

- Only representatives of the respective group discuss standardisation to minimise coordinating efforts

- Towards the mass integration, meeting were held weekly to ensure all teams are on the same page for the integration
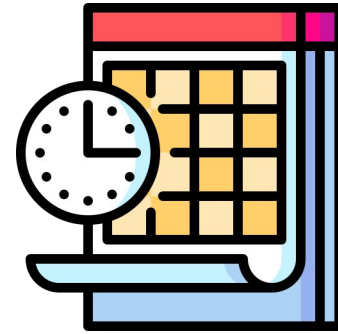
Challenges Faced

Time management

- Increase in complexity of team project
- School commitments such as tests, assignments and reports
  - Challenging to plan for team meetings
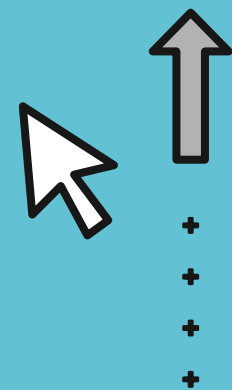- Unable to focus on team project fully

Overcome:

- Prioritize the work that make the most impact
  - Form builder and form elements
- Manage and communicate expectations clearly
  - Minimum viable product
- Defined goals, plans, responsibilities and expectations at the beginning

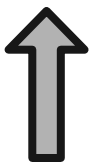08
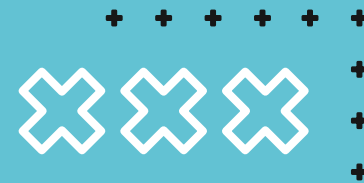
Future Work

Future Work

1. Additional Form Elements

There are still some form elements that have yet to be implemented.

For example:

- Rich Text Box
- Checkbox Grid
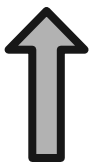- Multiple choice grid
- File Upload

Future Work

2. Assigning Forms to Specific Workers

- Currently, all form created will be available to all workers.

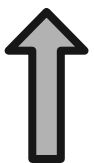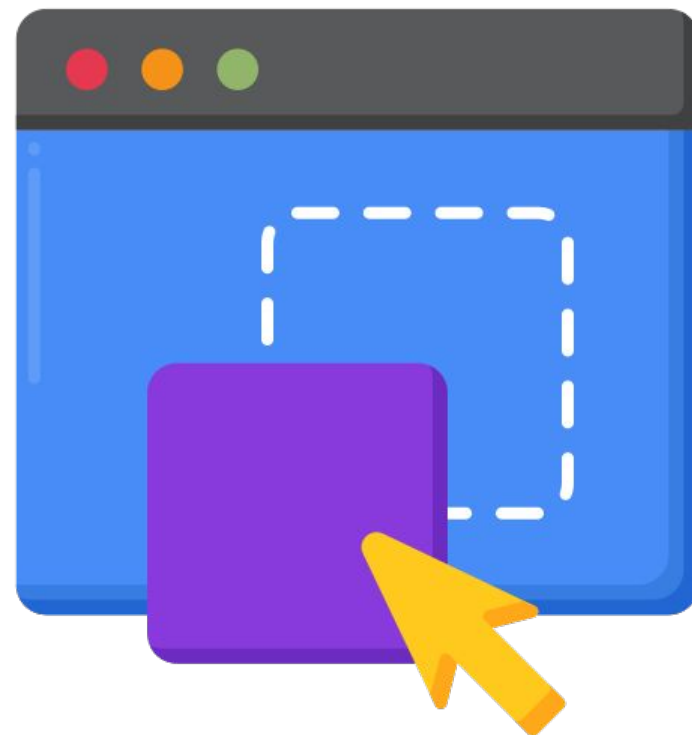- Propose that manager are able to assign forms to specific workers.

Future Work

3. Drag and Drop

- The current form builder is similar to google form
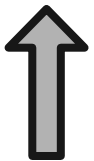
- Propose the form builder to be drag and drop format

Future Work

4. News Module

- Notify both manager and worker when there are new activity.

09

QnA

Thank you!