MCDA 5530 UI/UX Design & evaluation Group Project Presentation

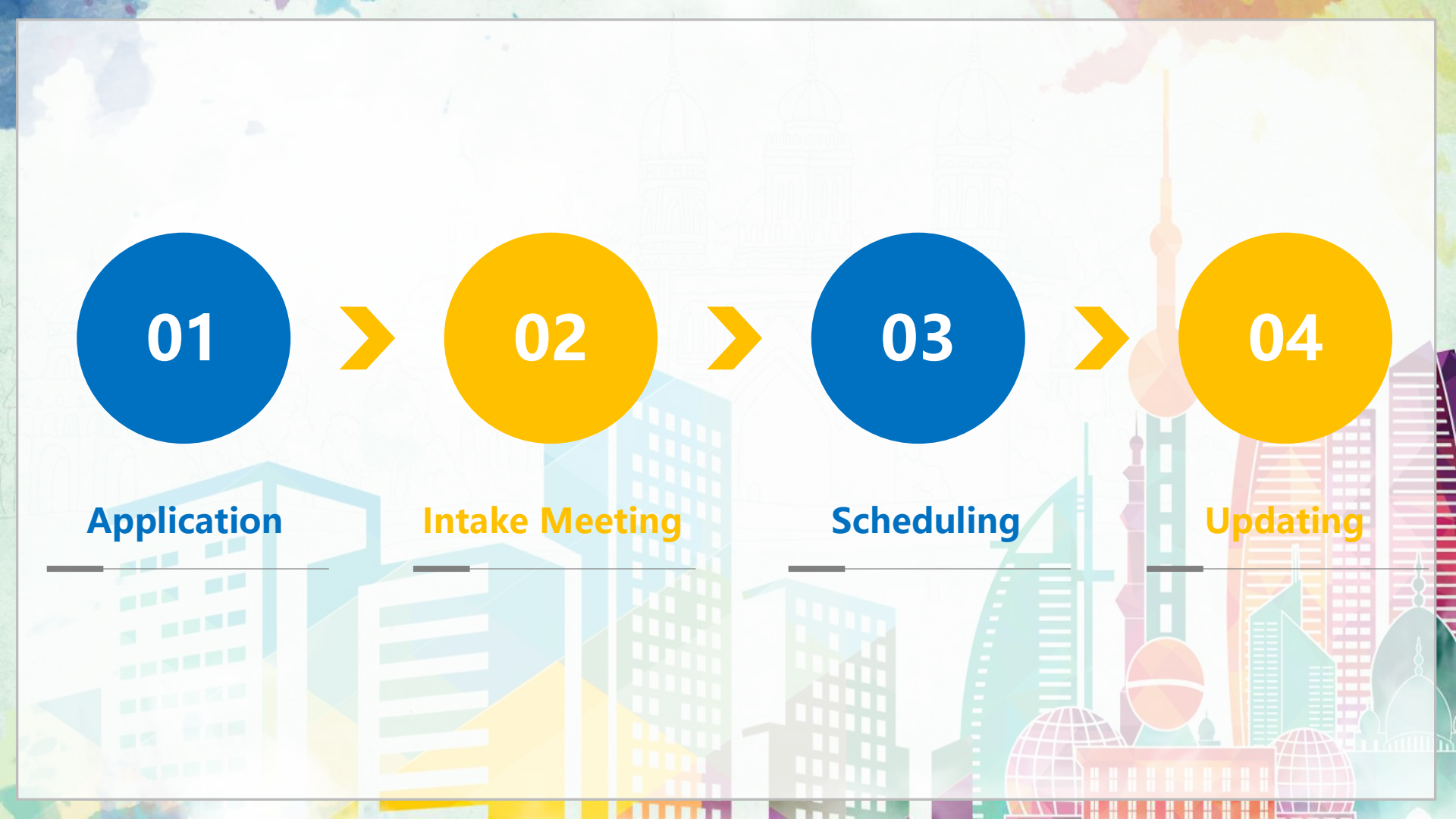
Present by: **Morounkeji Fasuyi, Madeleine Min Jing, Nikita Neveditsin Sadman Hoque Said, Bhagya Shree, Xinyun Tong**

CONTENTS

Part 1

Introduction

- Background
- Problem/Issue


01
02
03
04
Application
Intake Meeting
Scheduling
Updating

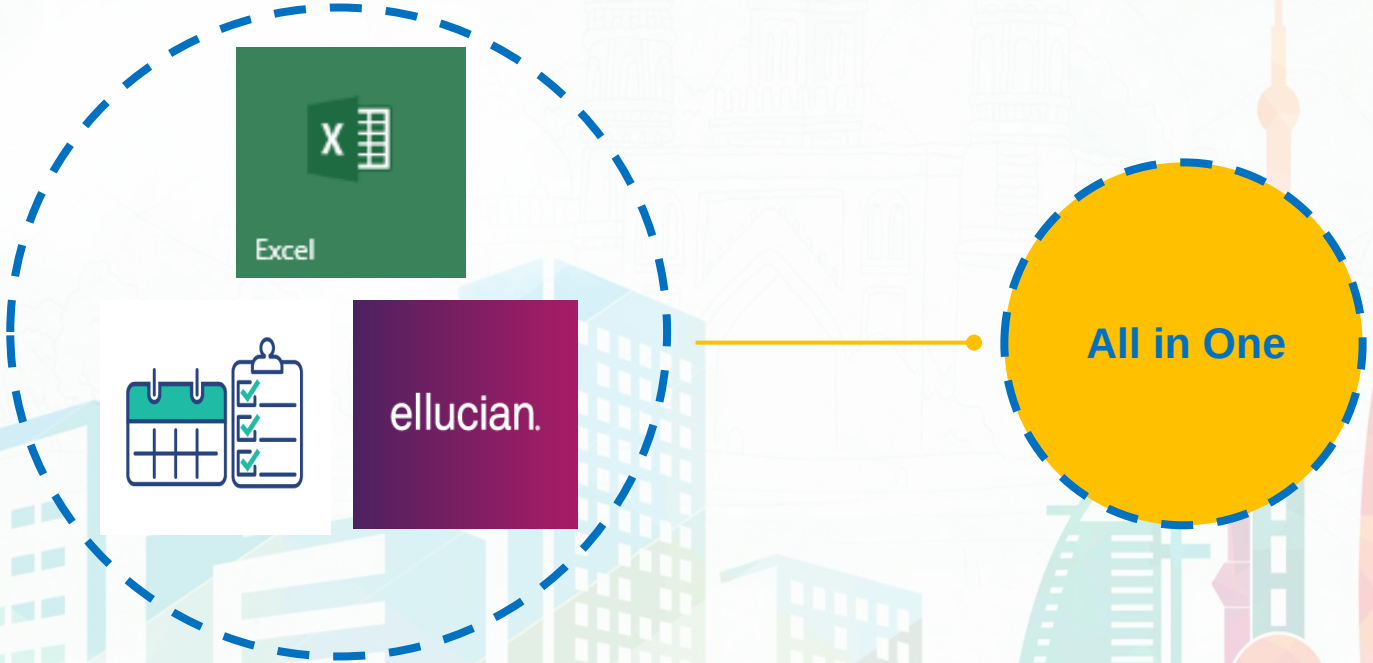
ellucian.

1 Student Information

Excel

2 Student Record

3 Department Schedule


Excel
ellucian.
All in One

Part 2

Methodologies

- Interview
- Observation
- Analysis
- Design Thinking

Methods of requirements gathering: Step 1

A meeting with employees of CSD

- Purpose: issues identification
- Date: October 12, 2018. Time: 2:30 PM
- Place: CSD
- Participants: two employees of CSD, 5 group members
- Core questions to employees of CSD: “
  - What kind of problems do students-clients of CSD have?”
  - “What difficulties do you face during your working day?
- Results: got two issues:
  - problem with navigation on campus (from students side)
  - problem with efficiency of tracking students status (from CSD side)

After long discussions and multiple voting processes the decision was made to work on the problem with efficiency of tracking students status. It was difficult to identify a persona for the navigation problem.

Methods of requirements gathering: Step 2A

A formal interview with employees of CSD

- Purpose: requirements gathering
- Date: October 26, 2018. Time: 2:00 PM
- Place: CSD
- Participants: an employee of CSD, 4 group members
- Core questions to employees of CSD:
  - “How frequently do you check the student banner for student schedule?”,
  - “What is the current procedure of maintaining information about students?”
  - “On an average, how many times do you check the banner?”
  - “What are the core things that you would like to see integrated?”
  - “What kind of information about students do you track?”
- Results: requirements collected for the product

Methods of requirements gathering: Step 2B

User observation

- Purpose: requirements gathering
- Date: October 26, 2018. Time: 2:30 PM
- Place: CSD
- Participants: an employee of CSD, 4 group members
- Question asked:
  - “Can you walk us through what you would do on the banner?”
- Results: an employee of CSD showed us what they usually do to get information about students and what kind of information it is.

User observation gave us better idea about user requirements

Gathered Requirements: overview

The solution should be able to:

- Provide CSD employees with daily alerts on student activities (e.g., course drop, not showed up etc.)
- Maintain a database with all current students who are registered with CSD
- Provide a user interface that allows to browse a list of all current students who are registered with Centre and show at least the following information:
  - Student’s phone number, address, e-mail address
  - List of courses
  - Schedule of classes
  - Schedule of exams
  - Special requirements
  - Emergency contact info
- The solution should interact with SMU banner system to get information about students from it

Requirements: Data Requirements

- Type of data: persistent database storage with daily backups
- Amount of data: up to 2 gigabytes/year
- Accuracy requirements: no special requirements as working with floating-point numbers is not necessary
- The data storage should be fault-tolerant (e.g., redundant database or 3rd party data storage provider)
- Data should be secured so that no 3rd parties have unauthorized access to it

Requirements: Environmental Requirements

- Physical
  - No special requirements. However, solution should be accessible in terms of interaction with students if there are such interactions
- Social: collaboration/corporation
  - Data is shared among multiple users. Race conditions/deadlock should be avoided while manipulating data
- Organizational
  - Should have an intuitive interface with no additional training required. Application support should be feasible by MCDA students
- Technical
  - Should work on any desktop machine with modern browsers
  - Internet connection should be provided
  - Should have access to SMU Banner system

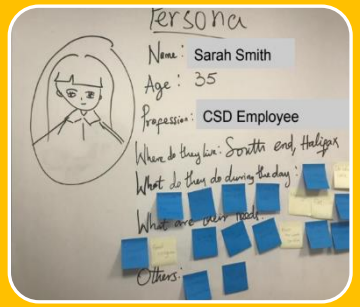
Requirements: User Requirements

- Intended user group: CSD employees
- Type of user: casual
- Age groups: mature specialists (20+ years old)
- Abilities:
  - Basic computer skills
  - Knowledge of CSD working environment and daily tasks
  - No previous experience is required
- Special needs:
  - Clients are students with disabilities
  - Might be used by future employees with disabilities

Requirements: Usability Requirements

- Reliability and responsiveness
- Field maintenance and serviceability
- Efficiency
- Intuitive design and learnability

Design thinking



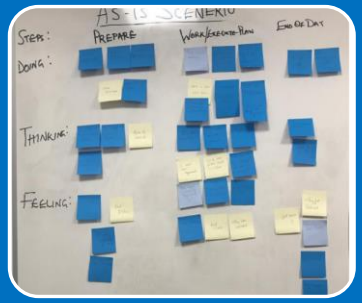
Persona

- Sarah Smith
- 35 years old
- Coordinator at CSD
- South End, Halifax
- Offer services to group of specially challenged students
- Access the information of her students efficiently

Empathy Path

- She says, thinks, does, and feels
- Pain point
  - The repetitive and inefficient means of tracking student course details
- Gain points
  - Concern for students keeps them motivated
  - She feels rewarded every time a student need is addressed

Design thinking



As-Is Scenario Map

- Prepare
  - Checks email, schedule, Think of students, highly motivated
- Work
  - Offer services, time consuming, irritated/frustrated
- End of day
  - Add remaining work to to-do list, work is inefficient, tired/good

Ideation

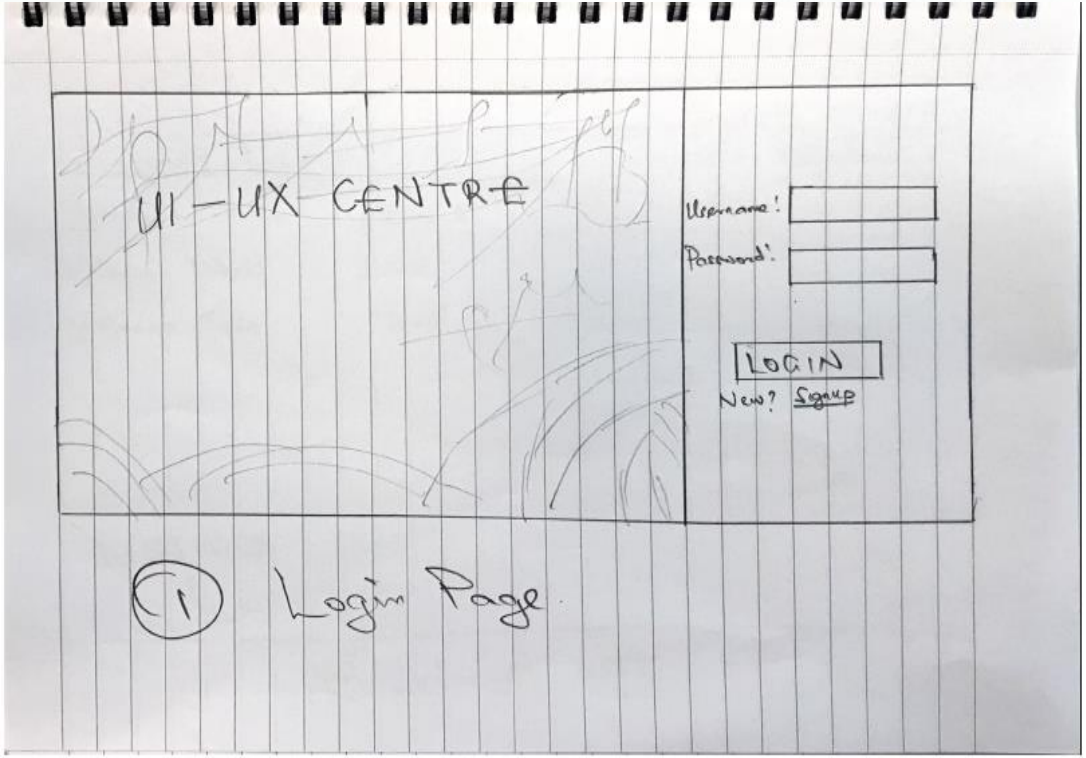
- Importance vs Feasibility graph
- No-brainers
  - Stand-alone application
  - Self service banner add-on

Part 3

Design Implementation

- Initial Sketch
- Heuristic Evaluation
- Updated Sketch
- Revised Sketch

Design Implementation: Initial Sketches

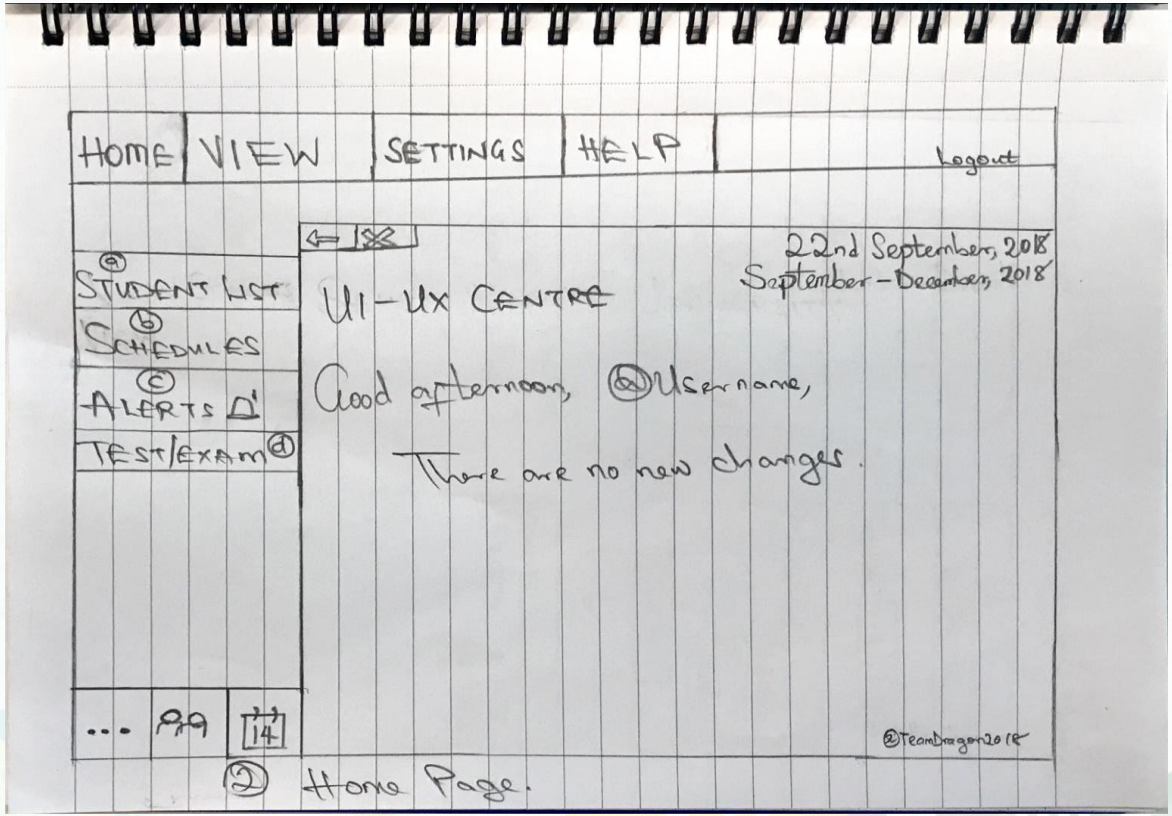


Heuristic Evaluation

‘Sign Up’ option is not displayed prominently; the text size is smaller

Forgot password option needs to be introduced.

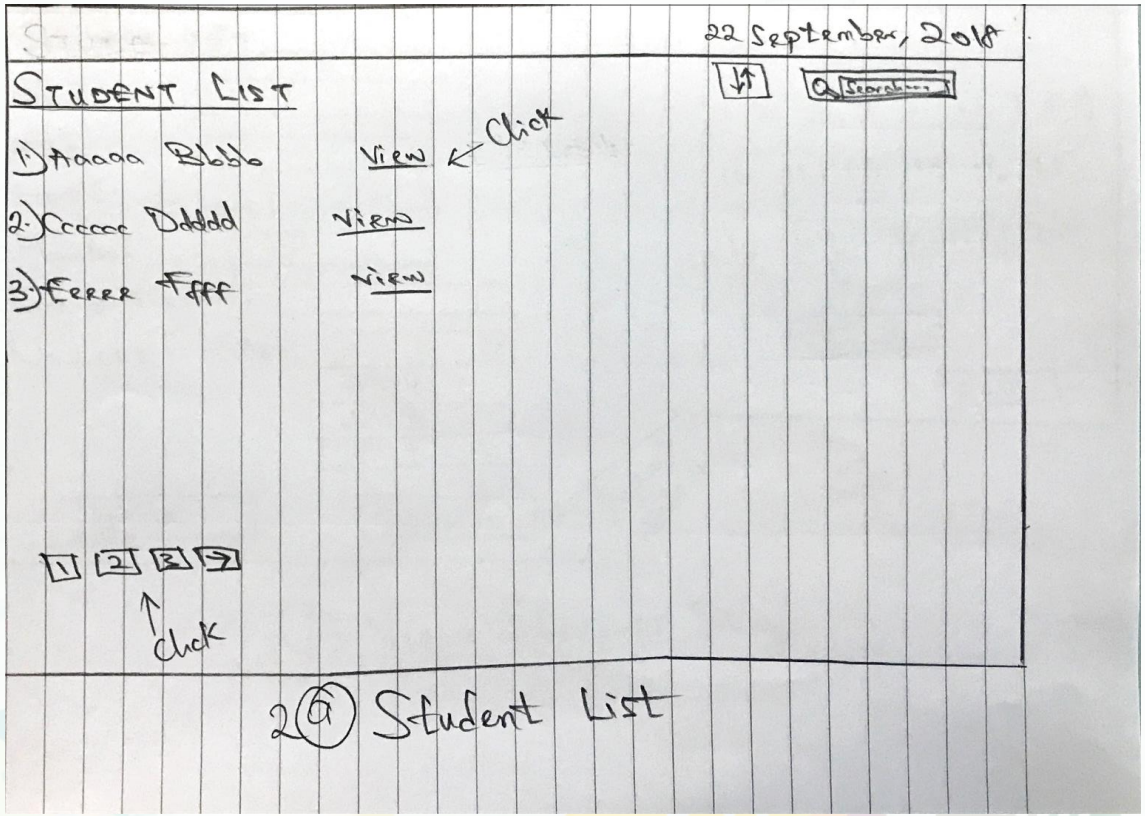
Design Implementation: Initial Sketches



Heuristic Evaluation

- Design is confusing and not minimalistic (horizontal and Vertical tabs)
- Semester and Date Information is not consistent throughout
- Tabs are closely located
- Footer is not readable.

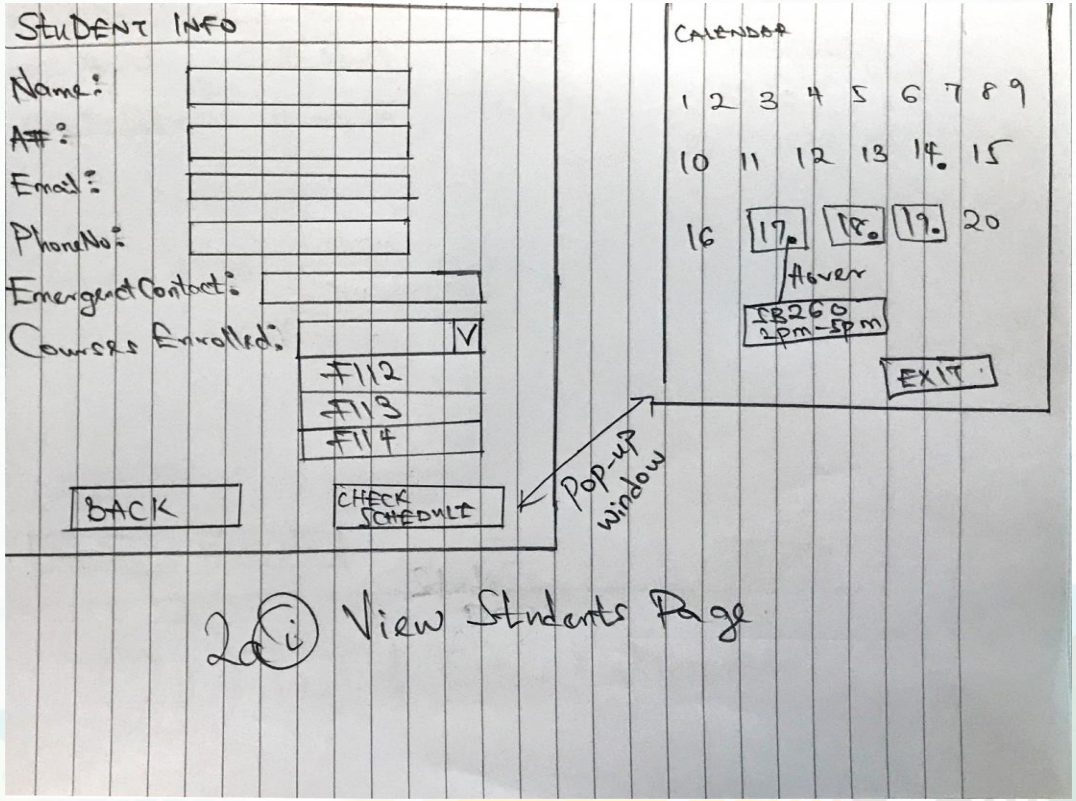
Design Implementation: Initial Sketches



Heuristic Evaluation

- Header bar is not consistent through out
- Back button is not consistent throughout
- Search bar on the student list
  not placed correctly and not visible enough.
- Too many pop ups for checking students .
- Logout button is not visible

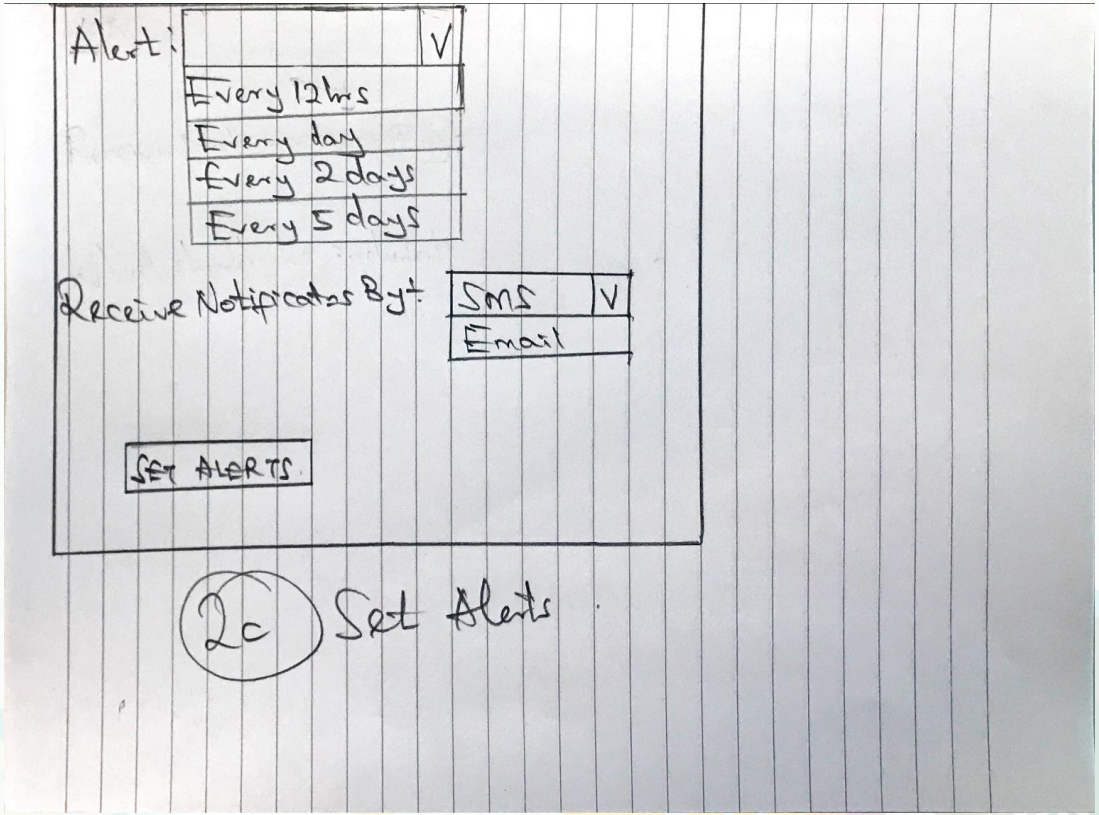
Design Implementation: Initial Sketches



Heuristic Evaluation

- Too many pop ups for checking schedule .(view-check schedule)
- Course enrolled field should not be dropdown ,it should be tabular form.
- Back button is not consistent

Design Implementation: Initial Sketches



Heuristic Evaluation

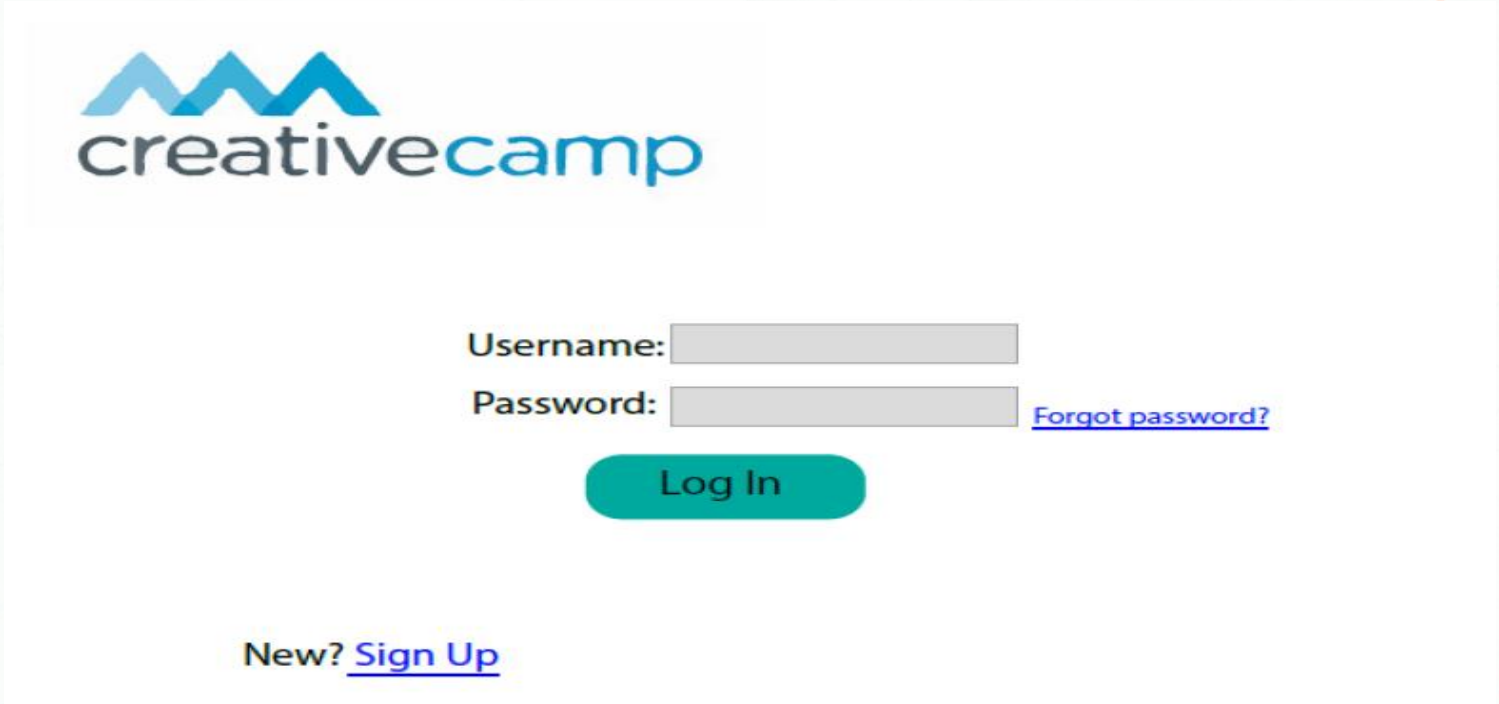
- Header bar is not consistent through out
- Back button is not visible
- Notification type is setting is not clear enough (like for what type of event ,we need notification)
- Alert type , is dropdown , radio button would be better.

Design Implementation: **Summary of Heuristic Evaluations**

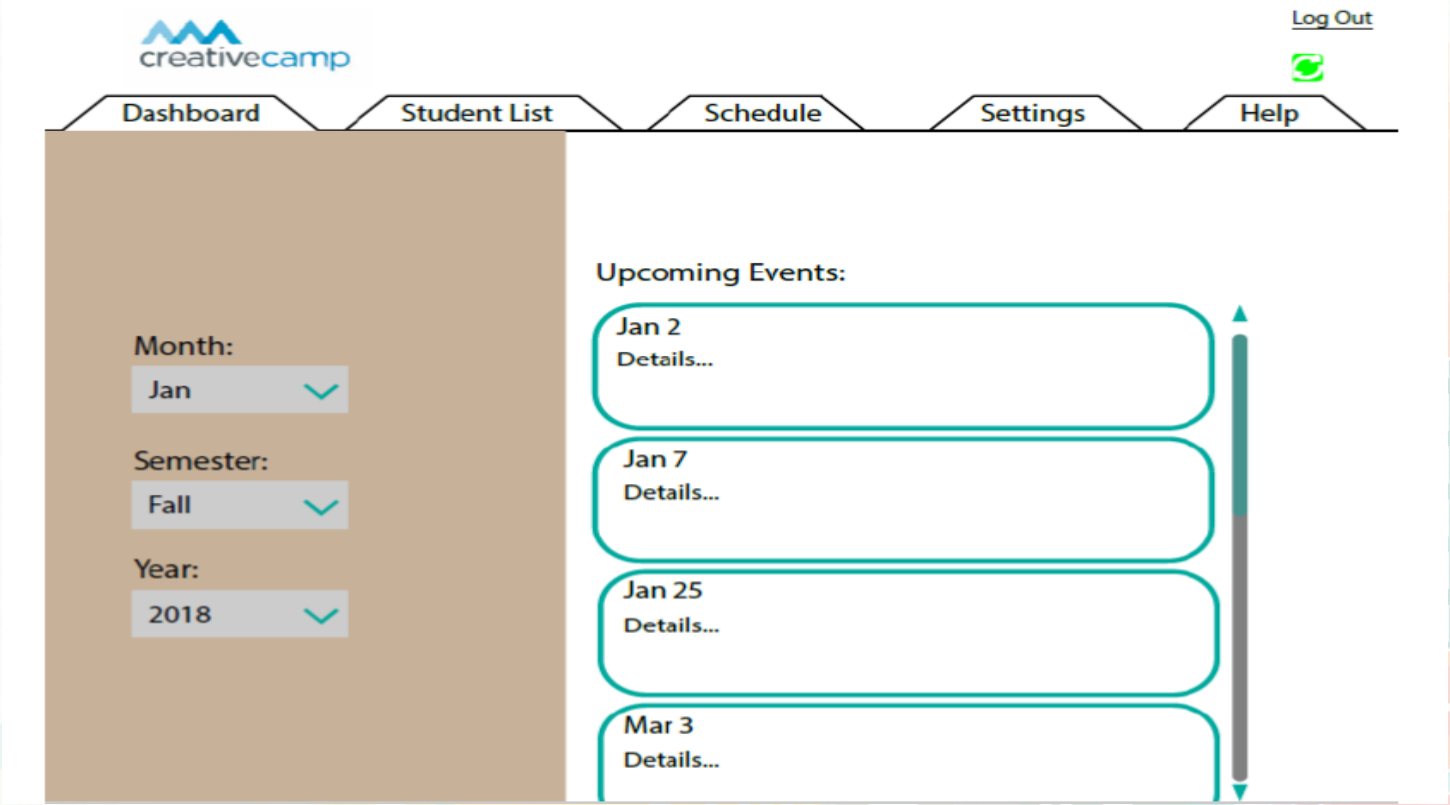
Heuristic evaluations helped to identify the following design flaws (summary):

- Design is too complicated: not minimalistic and not very intuitive
- Combination of horizontal and vertical tabs may be confusing
- Meaning of some buttons/links is hard to figure out
- No clear indication of where in the website the user is causing it difficult to understand how to go back to a specific page.
- Header bar and back button not consistent.
- Hard to navigate (e.g., hard to go to home page)
- Too many pop-up windows: for checking student details and schedule
- Lack of confirmation messages when deleting information
- Poor error handling

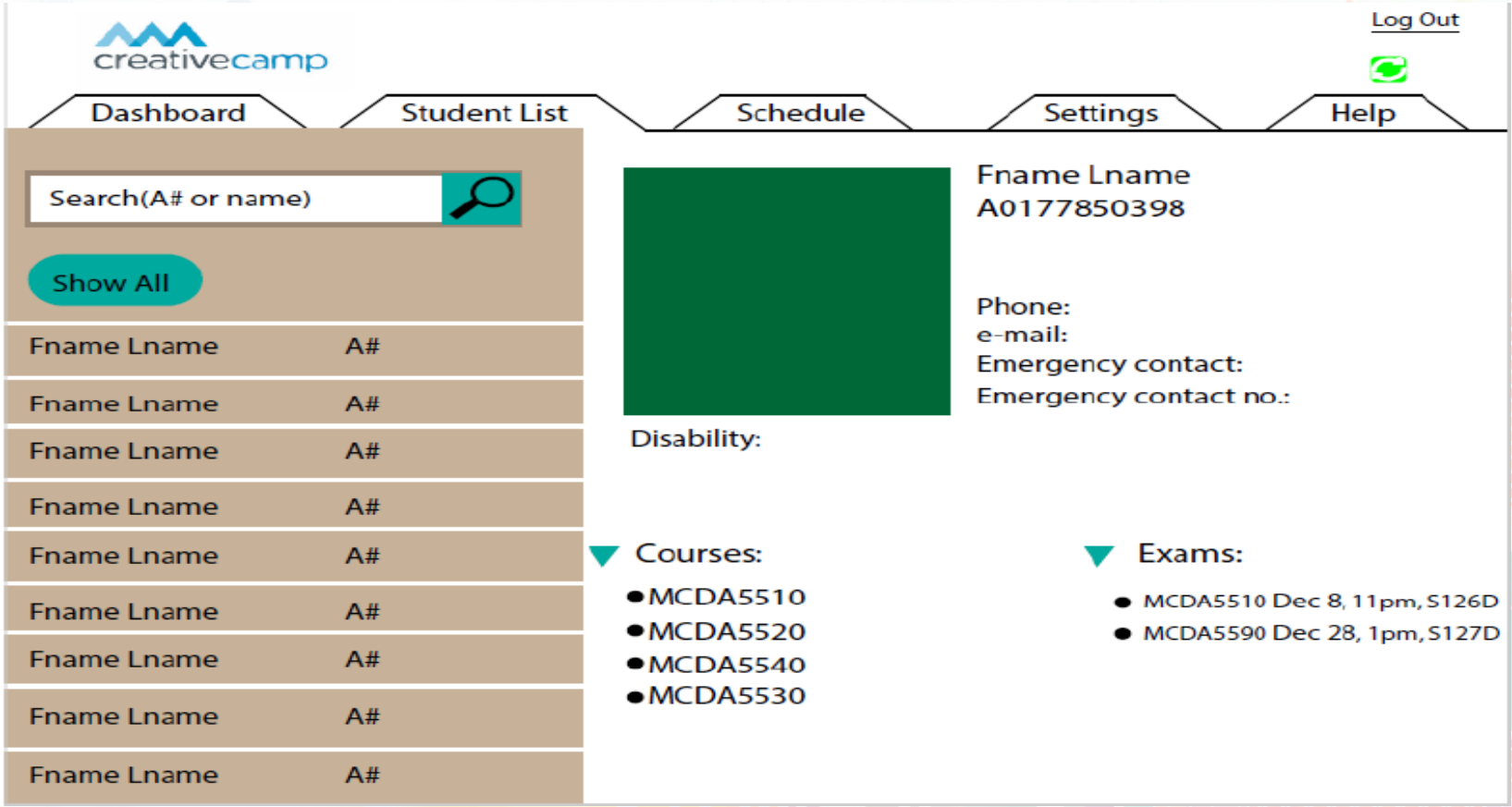
Design Implementation: Updated Sketches



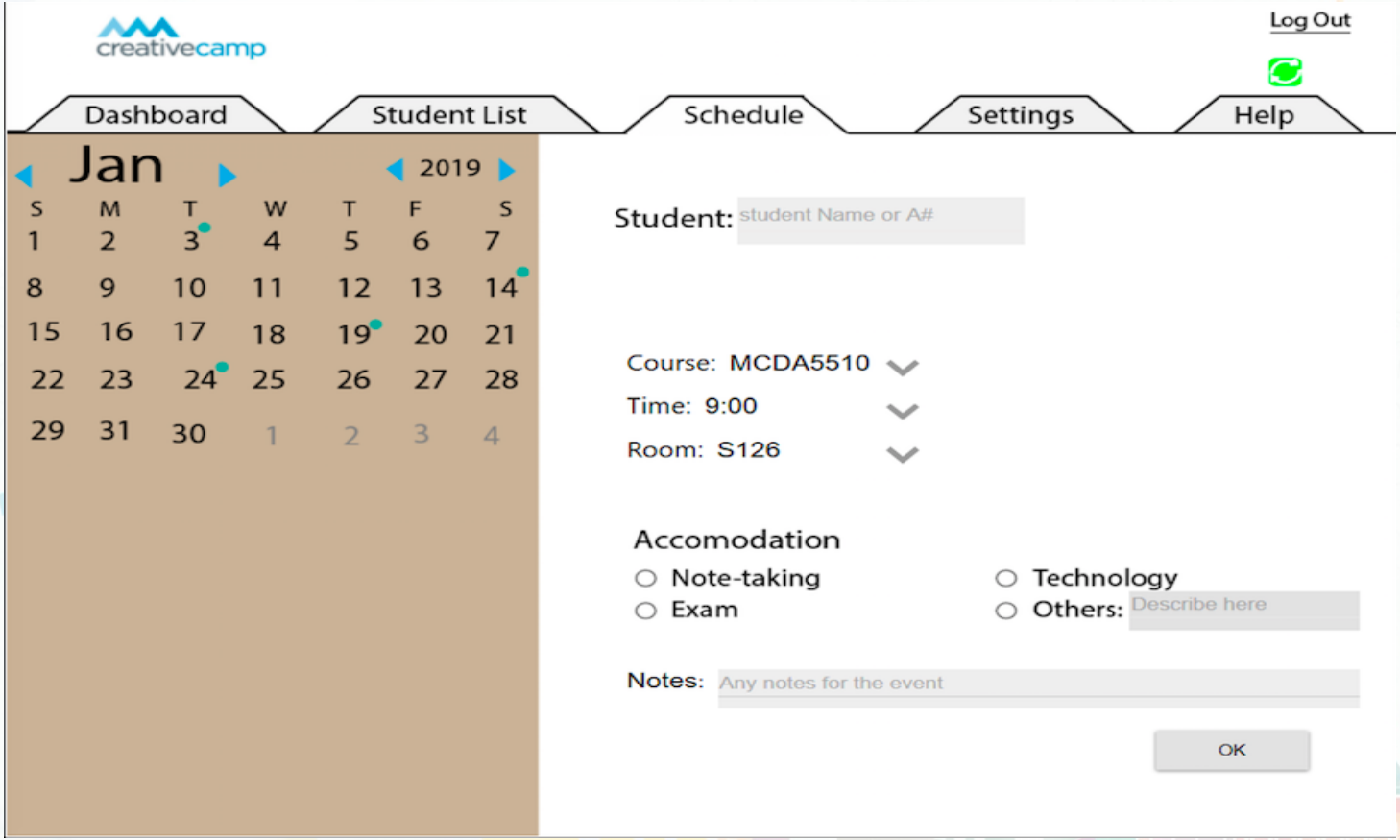
Design Implementation: Revised Sketches



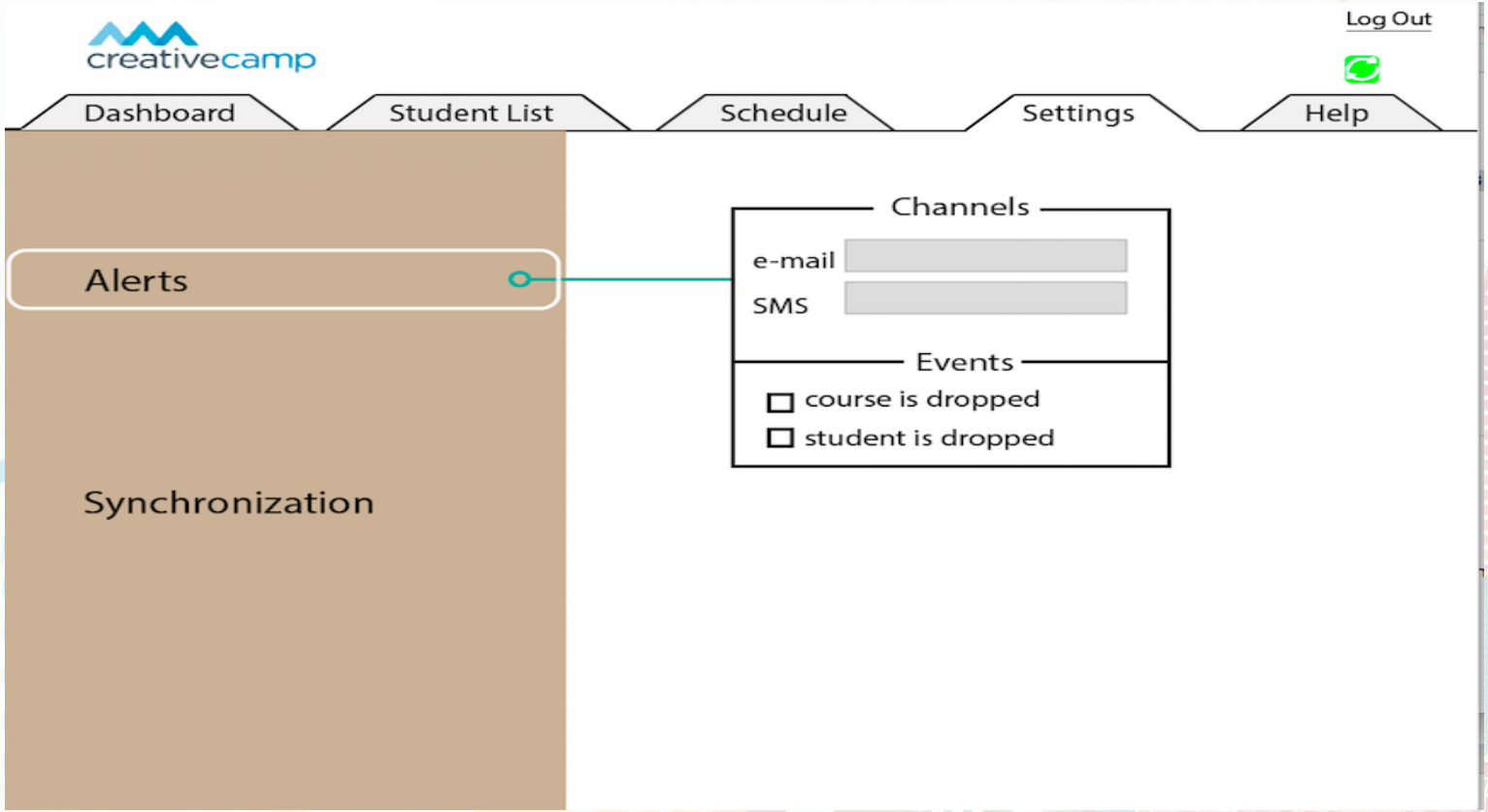
Design Implementation: Revised Sketches



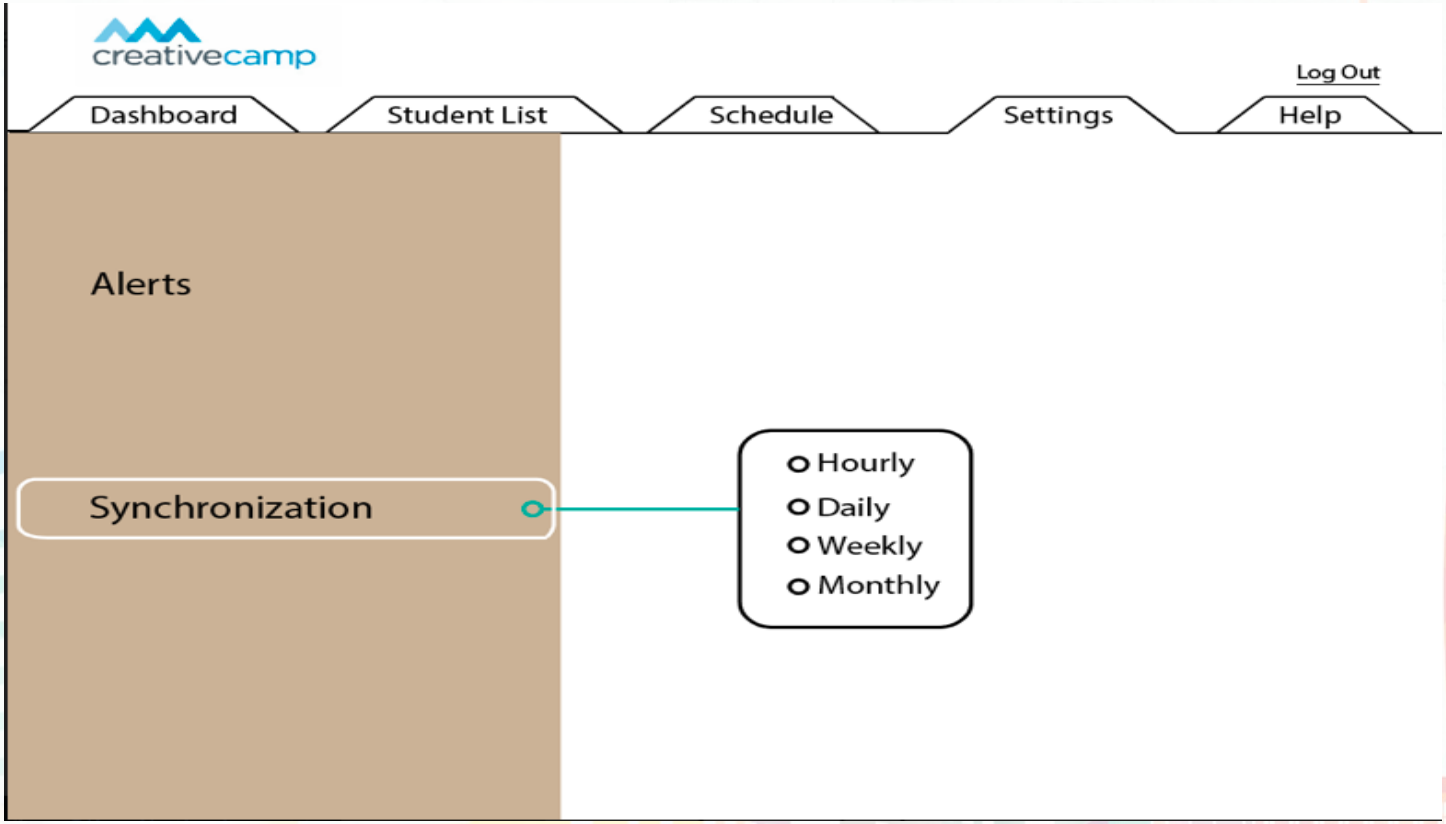
Design Implementation: Revised Sketches



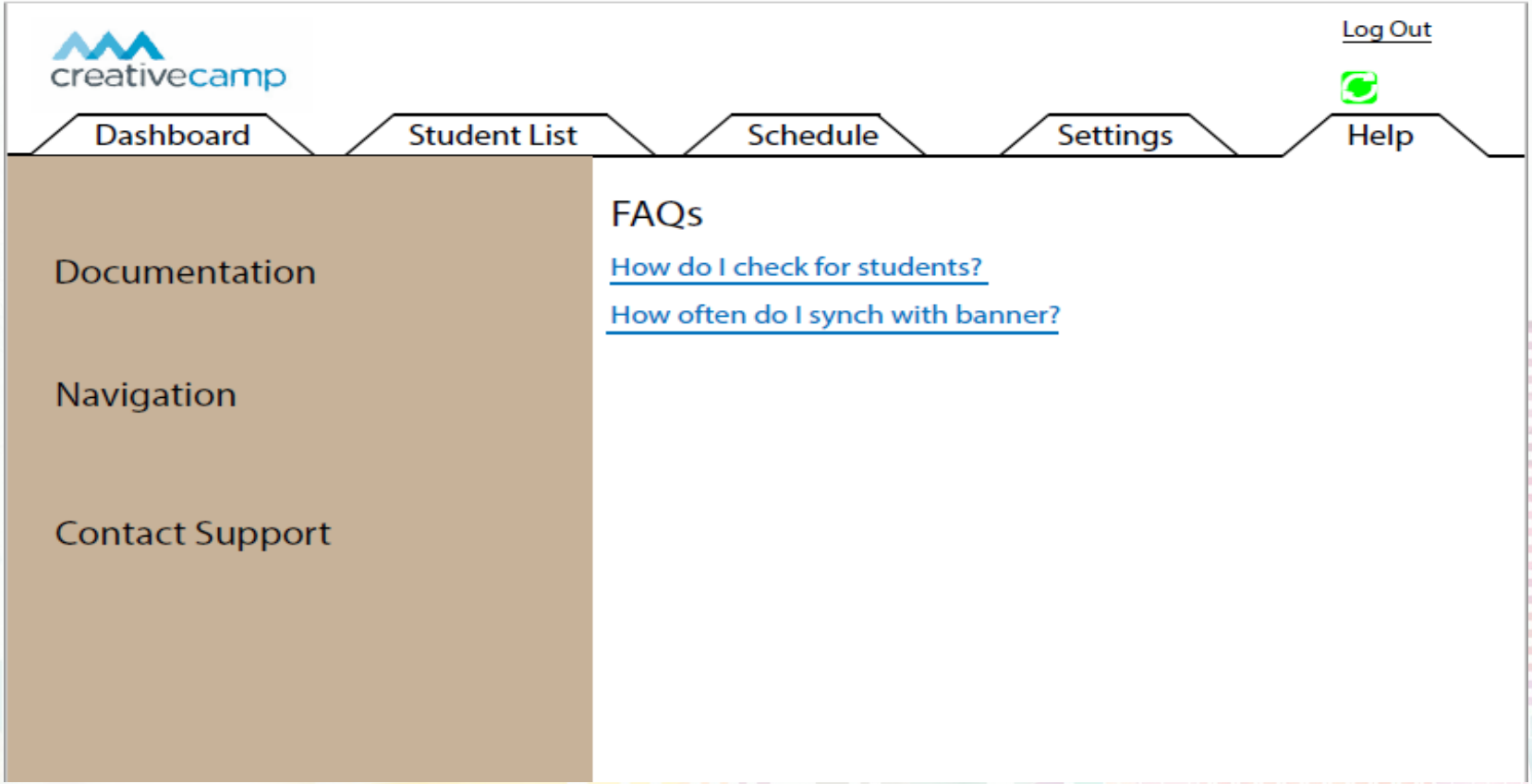
Design Implementation: Revised Sketches



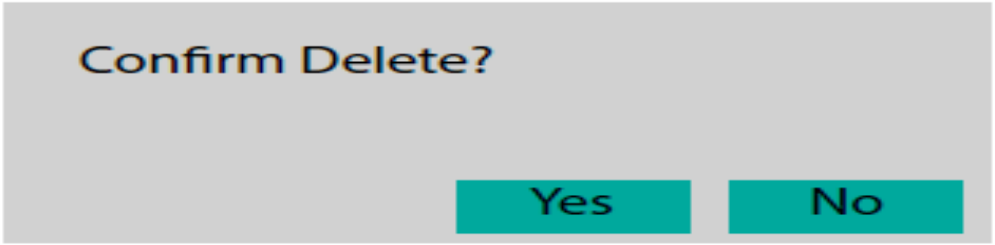
Design Implementation: Revised Sketches

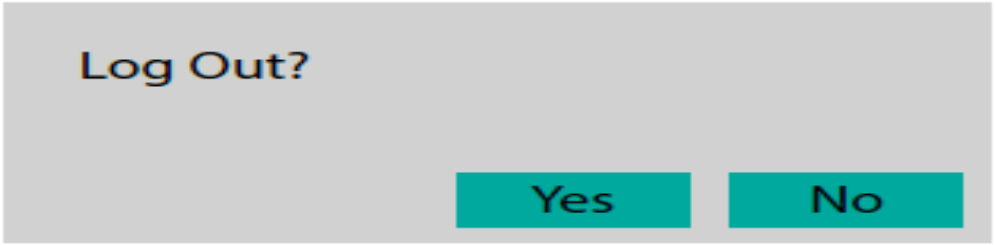


Design Implementation: Revised Sketches



Design Implementation: Revised Sketches





Part 4

Prototype Demo

- Video of Working Prototype

Part 5

Usability Studies

- Task List
- Steps
- Test Environment
- Test Procedure
- Questionnaire results
- Result and Evaluation

Usability Studies: Task List

1. Check if the proposed solution can help users of the application to get information about students faster than with the existing tools
2. Check if the proposed solution can help users to schedule some event for a student faster than with the existing tools
3. Make sure that the user interface is intuitive

Usability Studies - Steps

- To achieve the aforementioned tasks, we took into consideration 3 types of studies:
  - Prototype Testing
  - Questionnaire
  - Interview
- Number of participants: 3
- Summary of our UI
  - The UI that the user interacted with are the: Dashboard, Student List and Scheduling

Usability Studies – Test Environment

- Equipment used:
  - MacBook Air
  - QuickTime Player (Screen and voice recording)

Usability Studies: Test Procedure

- *Task 1 Question*: can a CSD employee get information about students using our application faster than with the existing tools?
- *Hypothesis*: a CSD employee should be able to get the information at least 2 times faster than with the existing tools
- *Variables*:
  - *Dependent*: use of our prototype VS use of the existing system
  - *Independent*: time
- *Procedure*:
- Prepare a laptop with screen recording software and open a prototype (team members)
- Ask a participant to perform these tasks on the laptop with recording software (team members):
  - Find a student in the application and get the needed information (courses, address, phone, etc.)
  - Delete the student from the system
- 3. Each participant follows the described steps (participants)
- 4. Ask some related questions (see interview transcript below) //in the end of all tasks
- 5. Propose to fill a questionnaire (see a questionnaire below) //in the end of all tasks

- **Task: Check if the proposed solution can help users to schedule some event for a student faster than with the existing tools**
- *Question*: can a CSD employee schedule some event for a student using our application faster than with the existing tools
- *Hypothesis*: a CSD employee should be able to schedule some event for a student at least 1.5 times faster than with the existing tools
- *Variables*:
  - *Dependent*: use of our prototype VS use of the existing system
  - *Independent*: time

- **Task: Make sure that the user interface is intuitive**

Question: is the user interface intuitive?

- *Hypothesis*: the user interface is intuitive and simple
- *Variables*:
  - *Dependent*: our user interface design
- *Independent*: intuitiveness of the UI
- *Procedure*:
- 4. Ask some related questions after performing Tasks 1 and 2 (see interview transcript below)
- 5. Propose to fill a questionnaire

UI/UX Evaluation Questionnaire

MCDA 5530

1. **How intuitive is the user interface of the application?**
   Mark only one oval.

	1	2	3	4	5	
not intuitive	◯	◯	◯	◯	◯	very intuitive

2. **How likely would you use this application to keep track of students with disabilities?**
   Mark only one oval.

	1	2	3	4	5	
unlikely	◯	◯	◯	◯	◯	very likely

3. **How likely would you use this application to maintain schedule of students' exams/special needs for note-taking, etc.**
   Mark only one oval.

	1	2	3	4	5	
unlikely	◯	◯	◯	◯	◯	very likely

4. **How likely would you use notification feature of this application (e.g. if some student drops a course, a notification on your email is sent)**
   Mark only one oval.

	1	2	3	4	5	
unlikely	◯	◯	◯	◯	◯	very likely

5. **Do you agree that this application can improve efficiency of daily activities in your department/company?**
   Mark only one oval.

	1	2	3	4	5	
disagree	◯	◯	◯	◯	◯	strongly agree

Example of User Feedback during testing

- Team Member: Please feel free to speak up as you go through the motions. We will like to hear your thought process as you go through the given scenarios.
- **Participant One**
- ***1st Scenario: Search for a Student named Tom Tong***
- \- So, I can search by First Name or Last Name? (*Response: Yes, you can)*
- \- Oh, I like how the picture comes up. Because it would be nice to have a face to go with the records.
- The first scenario is done. We will be recording the time it takes to perform the interaction.
- ***2nd Scenario: Delete the student.***
- \- Oh, so it's gone
- ***3rd Scenario: Schedule test/exam for Jan 26th***
- \- Okay so then to schedule an event, I'll probably go to schedule. Okay so this information is all for the same student? (*Response: Yes, it is*)
- \- Oh, that's nice
- ***4th Scenario: Delete the event***
- \- Okay do I edit for that? (*Response: No, there's a delete button)*
- \- Oh. Yes, there it is. The trashcan
- ***5th Scenario: View all events previously scheduled***
- \- Okay. Did we pass?

Usability studies: Questionnaire results

- Both participants gave 5 out of 5 marks for each of the question
- Open question answers:
  - P1: *I really like the simplicity of the application. Having a snapshot of each student is very useful*
  - P2: *Interface and Banner are very important but prototype is very promising*

Sample Questions and Answers

- ***Q: Do you think our prototype will improve on your system?***
- I think so. It has all of the information on one spot. We have to login to banner and go through all of those steps instead of having it right there. And having this interface, all of the information we're looking for is going to be available there. It's a time saver for us for sure.
- ***Q: Do you find that our prototype for scheduling is more convenient than Excel?***
- Oh yes, we have all of the information there so that would save a lot time of going back and forth.
- Also, if we had a better idea of the potential number of students that could be writing here?
- Because often one of the things that we do get asked is a professor just wants to know "Oh how many people do I have writing over there?" They just want the total number. There no way to catch all the students that we have that are currently registered in a particular course. Like maybe a search engine like how many people are in FIN2350?

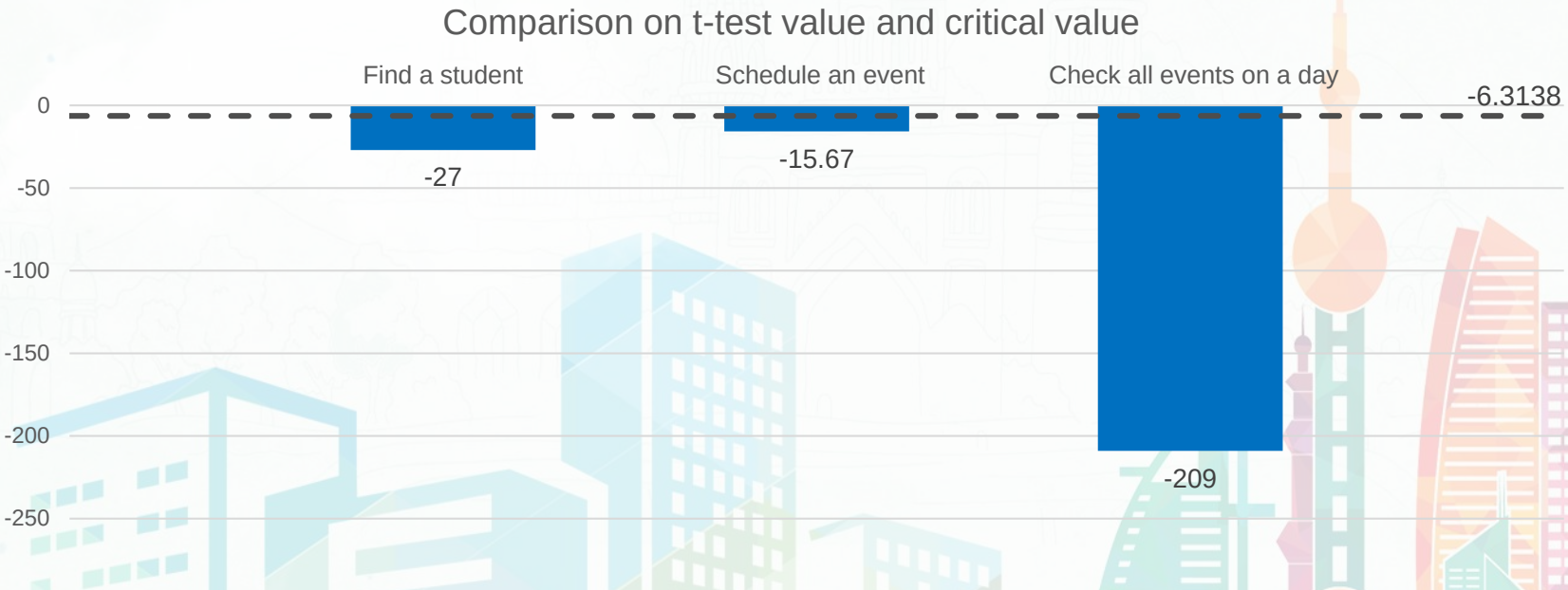
Usability Study: Result and Evaluation

- Hypothesis Testing
  - Null Hypothesis: There is no time difference of performing the task ($\mu_d \geq 0$)
  - Alternative Hypothesis: The time difference is less than zero ($\mu_d < 0$)
  - *at alpha level of **0.05***
- Critical Value is **-6.3138** (degree of freedom = 1)

Usability Study: Result and Evaluation

Task	Participant	Time Taken (Existing Application)	Time Taken (New Application)	Difference	Mean	Standard Deviation
1	1	90	25	-65		
Find a student	2	90	20	-70	**-67.50**	**3.54**
2.1	1	150	25	-125		
Schedule an event	2	150	40	-110	**-117.50**	**-15.67**
2.2	1	120	15	-105		
Check all event	2	120	16	-104	**-104.5**	**0.71**

Usability Study: Result and Evaluation



- **t-test value < critical value**
- **reject null hypothesis**
- **We can conclude that it is taking less time to perform these tasks in the new application.**

Part 6

Future Work

- Keep Information
- Implement Resource Management

Future work

The following additional features may be useful:

- Keeping information about students for at least 5 years. Mark students as “active” and “inactive” in the system
- Implement resource management, e.g. define a set of rooms available, staff members available and use this information for scheduling activities (For example, if room SB111 is used on Friday, December 28 from 11:30 to 12:30, then users of the application should not be able to schedule another event in this room within this time frame)


Thank You!