

Schedulity Screen Sketches

By AN 2

Elizabeth Schmitt

Robert Stickler

Benjamin Towle

Wesley Jansen

Our project name is Schedulity. This is a scheduling application that has a focus on a student, their courses, and study time. We have identified four actor groups. Each of these people have a separate role in the scheduling process.

The first actor is Administrator. This person basically has control over every user and their individual calendar. This person can delete users and edit users. They can also delete, edit, and move a user's classes and activities. This person is able to edit everything in case some problem comes up with the users account or courses. The administrator has access to all the tables in case they need editing.

The second actor is the advisor. This person has the power of an Iowa State advisor. Every advisor has one or more students who's classes they can edit. Once the student has input their individual courses the advisor is the one that can drop, switch, and edit that course. Advisor and administrator are the only ones that can edit the time, location, and deletion of courses of a student once that student has entered that course into their calendar. We will try to give this person the ability to chat with their students so that they can figure that all out within the application instead of having to leave in order to email a student. So each advisor is connected to their students and has the ability to see only their classes not the rest of their activities and study time that are on the students calendar as well.

The next actor is the student. This is the main user of our application and the person who gets the most use out of Schedulity. All the other actors are oriented towards helping the students. This person first enters their courses into the application along with all the courses information and how important this course is to them. Once they enter how much time they want to spend studying every week an amount of study time will be given to each course based on that courses importance to the person. They then enter other activities that they have weekly, when they sleep, and time they spend at the gym. Once these things are entered into the system a calendar is created that helps the student to organize their week and manage their time. Students can also connect with other students and become friends. They can see what classes they share with their friends. Friends can also set up study or hangout times with their friends which they can share on their calendars. Friends can look at each other's calendars which is very convenient for setting up times to meet or study. The student can edit their classes only by going through their advisor who can edit what classes they are in when. Friends can also message each other to set up meeting times and discuss courses. This person has the power to edit and add activities, gym, and study times for themselves. The student can also give their parent abilities to see and comment on their schedule.

The last actor in schedulity is the parent. This person is the parent of a student that uses the application. A parent can have one student on this application and they only have viewing ability. The parent can view their students schedule to see what they are doing that week. The parent also has the ability to make suggestions on their student's schedule. So if the parent thinks their student should spend their time doing more or less time studying, for example, they can create a comment indicating this. The student must invite their parent to be able to see and comment on their schedule.

Non-Functional Requirements

1. Usability

Our app should be easy and intuitive to use. Good usability can enhance user satisfaction and encourage them to continue using the app.

2. Performance

Our App should be designed to perform quickly with minimal to make a better user experience. A well performing app can also save battery power.

3. Security

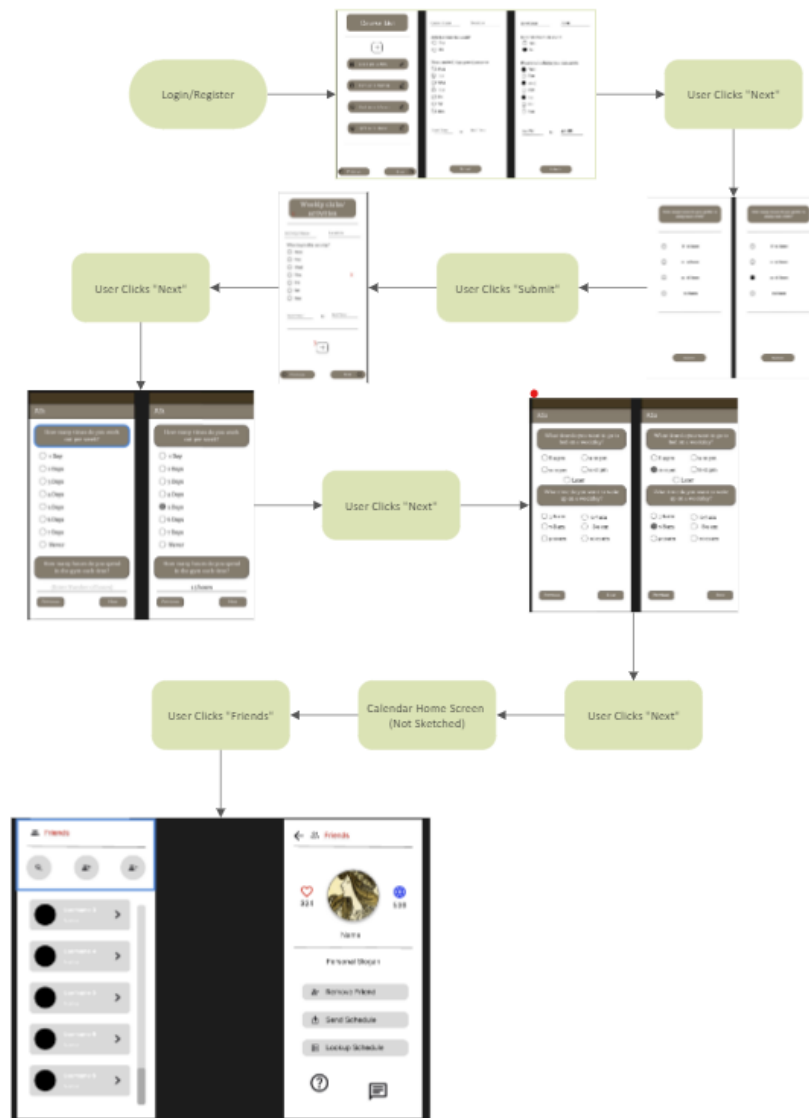
Because we are storing personal data, we are required to have good security to avoid data breaches.

4. Access Our app should be able to support various users (actors) such as the parents, admin, and advisors in addition to students to allow changes to be made to student's schedules depending on which actor is in use

List tables and fields

- User: Normal User that has classes, activities, friends, a parent and an advisor
 - Id
 - Username
 - Password
 - Email
 - Logged In?
 - Courses (Many to Many)
 - Friends (Many to Many)
 - Advisor (Many to One)
 - Parent (One to One?)
 - Other Activities(One to many?)
- Courses: courses and how they operate and when they are and details
 - id
 - Course
 - Location
 - Days
 - StartTime
 - EndTime
 - Importance
 - Credits
 - Lab
 - Time
 - Users (many to many)
- Friends: Friends can view your calendar to see free time and such
 - Id
 - Username
 - Email
 - Logged in?
 - Users (many to many)
- Parent: can veiw a users thing and make suggestions
 - Id
 - Username
 - Password
 - Email
 - Logged in?
 - User (one to one)
- Advisor: can drop courses for user and edit courses
 - Id
 - Email
 - Username
 - Password

- Users (one to many)
- Other Activities: extracurriculars that a user can be in
 - Users (many to One)
 - Name
 - Time
 - Start
 - End
 - Days
 - Location
- Admin: can change everything at any time has access to any table with full editing abilities
 - Id
 - Username
 - Password
 - email



The image shows two side-by-side mobile app screens for a 'Study Time Screen'. Both screens are titled 'How many hours do you prefer to study each week?'. The left screen (iPhone 14 - 4) has a red '1' next to the title and a red '2' next to the first radio button. The right screen (iPhone 14 - 5) has a black radio button selected for '14 - 16 hours'. Both screens have a 'Submit' button at the bottom, with a red '3' next to it on the left screen.

Name: Study Time Screen

Developed by: Elizabeth Schmitt

Purpose: To understand how long the student wants to study in a week so that Schedulity can generate. The time spent on each class will then be calculated and output for the user to see and adjust.

Interface items:

1. This item prompts the user as to how many hours they would prefer to study in one week.
2. Here the student chooses how many hours they want to spend studying in the week. This time affects the users overall schedule with how much space they will have in their schedule to do non-academic things.
3. When the user pressed this button, an algorithm is done using the hours indicated. This algorithm takes the users courses and looks at their importance. Based on these numbers given for importance the algorithm takes the total study time indicated and finds a time to spend studying each course based on importance. This creates a schedule for the student with the given study times.

The image displays two wireframe screens for a mobile application, labeled 'Frame 2' and 'Frame 6'.

Frame 2: This screen is titled 'Weekly clubs/ activities' (indicated by a red '1'). It contains the following elements:

- Input fields for 'Activity Name' and 'Location'.
- A section titled 'What day is this activity?' with radio button options for Mon, Tue, Wed, Thu, Fri, Sat, and Sun. A red '2' is placed next to the 'Thu' option.
- Input fields for 'Start Time' and 'End Time' separated by a 'to' label.
- A plus sign button (+) with a red '3' next to it.
- Navigation buttons at the bottom: 'Previous' and 'Next'.

Frame 6: This screen shows a filled-out version of the form. It is titled 'Weekly clubs/ activities'.

- The 'Activity Name' field is filled with 'Digital Women'.
- The 'Location' field is filled with 'Coover'.
- The 'What day is this activity?' section has the 'Thu' option selected with a black dot.
- The 'Start Time' field is filled with '6:30' and the 'End Time' field is filled with '7:30'.
- The plus sign button (+) is present at the bottom.
- The navigation buttons 'Previous' and 'Next' are at the bottom.

Name: Activities Screen

Developed by: Elizabeth Schmitt

Purpose: To add activities that the student has weekly to their schedule. This will help in designing the overall calendar for the user. By inputting these events it will create a calendar with the users prioritize.

Interface items:

1. This item prompts the student to enter they clubs/ activities that the user has every week.
2. The user then enters the name of the activity whether it be work, a club, or something that isn't school related. Then the location of the activity is input. The student selects the day and time of this activity so that it can then be added to their calendar for every week at the same time.
3. This plus symbol actually adds the input activity to the data base that stores all the other activities besides the courses. This determines what is on the schedule that is created for the person. The input activity is connected to the user that added the screen.

The image displays two mobile app screens side-by-side, labeled 'iPhone 14 - 2' and 'iPhone 14 - 3'. Both screens have a header 'AS1' and a title bar.

Screen 1 (iPhone 14 - 2):

- Question 1: "How many times do you work out per week?" with a red "1" next to it.
- Options: Radio buttons for "1 Day", "2 Days", "3 Days", "4 Days", "5 Days", "6 Days", "7 Days", and "Never".
- Question 2: "How many hours do you spend in the gym each time?" with a red "2" next to it.
- Input field: "(Enter Number of hours)".
- Buttons: "Previous" and "Next" (with a red "3" next to it).

Screen 2 (iPhone 14 - 3):

- Question 1: "How many times do you work out per week?"
- Options: Radio buttons for "1 Day", "2 Days", "3 Days", "4 Days", "5 Days" (selected), "6 Days", "7 Days", and "Never".
- Question 2: "How many hours do you spend in the gym each time?"
- Input field: "1.5 hours".
- Buttons: "Previous" and "Next".

Name: Workout Screen

Developed by: Wesley Jansen

Purpose: To know how much a person wants to work out a week and how long it takes them to take into account what they need to have time for throughout any given week. This will help generate the recommended calendar for the user and one step closer.

Interface items:

1. This item prompts the user to enter the number of days they work out a week
2. Then also prompts the user to enter the number of hours they work out every time they work out.
3. Then they can move onto the next part of the user journey by pressing next and previous question by pressing previous

The image displays two side-by-side mobile app screens, labeled 'iPhone 14 - 6' and 'iPhone 14 - 7', both titled 'AS2'. Each screen contains two sections for selecting times on weekdays.

Screen 6 (Left):

- Section 1:** 'What time do you want to go to bed on a weekday?' with a red '1' next to the question. Options: ☐ 8-9 pm, ☐ 9-10 pm, ☐ 10-11 pm, ☐ 11-12 pm, and ☐ Later.
- Section 2:** 'What time do you want to wake up on a weekday?' with a red '2' next to the question. Options: ☐ 5-6 am, ☐ 6-7 am, ☐ 7-8 am, ☐ 8-9 am, ☐ 9-10 am, and ☐ 10-11 am.
- Navigation:** 'Previous' and 'Next' buttons at the bottom, with a red '3' between them.

Screen 7 (Right):

- Section 1:** 'What time do you want to go to bed on a weekday?'. Options: ☐ 8-9 pm, ☐ 9-10 pm, ☒ 10-11 pm, ☐ 11-12 pm, and ☐ Later.
- Section 2:** 'What time do you want to wake up on a weekday?'. Options: ☐ 5-6 am, ☐ 6-7 am, ☒ 7-8 am, ☐ 8-9 am, ☐ 9-10 am, and ☐ 10-11 am.
- Navigation:** 'Previous' and 'Next' buttons at the bottom.

Name: Bed Time

Developed by: Wesley Jansen

Purpose: Another step of the user journey to figure out what time a user wants to go to bed for the weekdays

Interface items:

1. Questions when you wish to go to bed on a weekday
2. Questions when you wish to wake up on weekdays
3. Gives you the next and previous options for the user journey

The image displays two side-by-side mobile app screens, labeled 'iPhone 14 - 8' and 'iPhone 14 - 9', both titled 'AS3'. The screens are part of a user journey for selecting weekend schedules.

Screen 1 (iPhone 14 - 8):

- Header: AS3
- Question: "What time do you want to go to bed on a weekend?" with a red number '1'.
- Options:
 - ☐ 8-9 pm
 - ☐ 9-10 pm
 - ☐ 10-11 pm
 - ☐ 11-12 pm
 - ☐ 12-1 am
 - ☐ 1-2 am
- Question: "What time do you want to wake up on a weekend?" with a red number '2'.
- Options:
 - ☐ 5-6 am
 - ☐ 6-7 am
 - ☐ 7-8 am
 - ☐ 8-9 am
 - ☐ 9-10 am
 - ☐ 10-11 am
 - ☐ 11-12 am
- Navigation: "Previous" button, a red number '3', and a "Next" button.

Screen 2 (iPhone 14 - 9):

- Header: AS3
- Question: "What time do you want to go to bed on a weekend?"
- Options:
 - ☐ 8-9 pm
 - ☐ 9-10 pm
 - ☐ 10-11 pm
 - ☐ 11-12 pm
 - ☒ 12-1 am
 - ☐ 1-2 am
- Question: "What time do you want to wake up on a weekend?"
- Options:
 - ☐ 5-6 am
 - ☐ 6-7 am
 - ☐ 7-8 am
 - ☐ 8-9 am
 - ☒ 9-10 am
 - ☐ 10-11 am
 - ☐ 11-12 am
- Navigation: "Previous" button and a "Next" button.

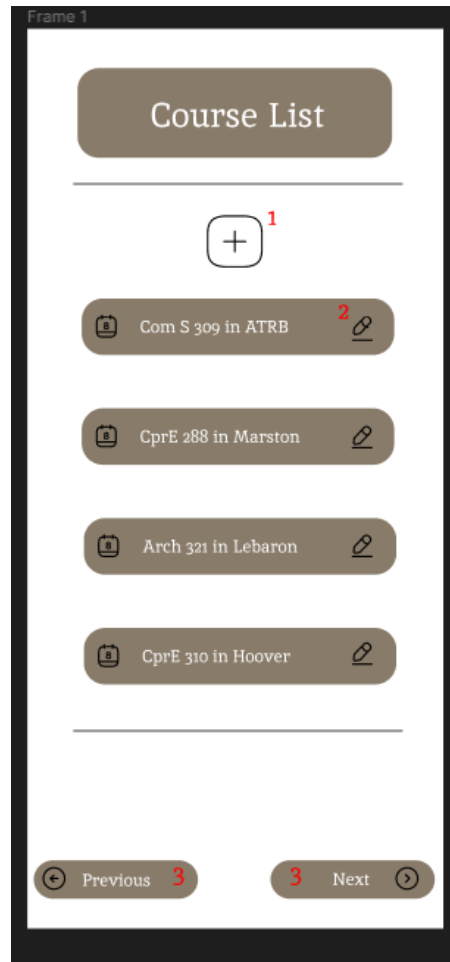
Name: Weekending

Developed by: Wesley Jansen

Purpose: Another step of the user journey to figure out what time a user wants to go to bed for the weekend

Interface items:

1. Questions when you wish to go to bed on the weekend (goes as late as 1-2 am because if you stay up later than that then its just silly.)
2. Questions when you wish to wake up on the weekend (goes as late as 11-12 am because you should be up in the morning).
3. A next and previous button to go between questions along the user journey.



Name: Course Input Screens

Developed By: Benjamin Towle

Purpose: To allow the user to add, edit, and delete courses as they wish. This screen is a part of the entire user journey process.

Interface Items:

1. "+" Button prompts the user to add a course to their schedule
2. Edit button next to the course name and location prompts the user to edit one of the classes they have already added
3. Next and previous buttons allow the user to switch between screens of the user journey

Frame 7

Course Name ² Location

Does this class have a lab?

¹ ☐ Yes

☐ No

Please mark all days your class meets

☐ Mon

☐ Tue

☐ Wed

³ ☐ Thu

☐ Fri

☐ Sat

☐ Sun

Start Time to End Time

⁴ Submit

Frame 8

Com S 309 ATRB

Does this class have a lab?

☐ Yes

☒ No

Please mark all days your class meets

☒ Mon

☐ Tue

☒ Wed

☐ Thu

☒ Fri

☐ Sat

☐ Sun

3:20 PM to 4:10 PM

Submit

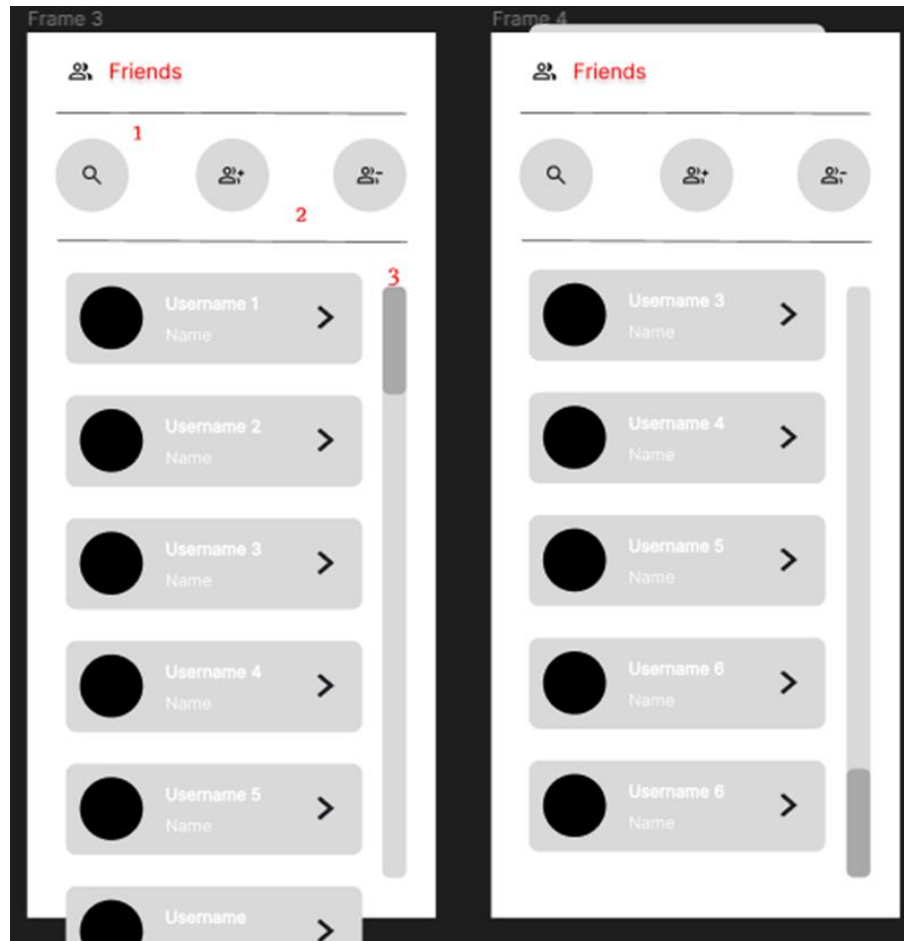
Name: Course Details Screens

Developed By: Benjamin Towle

Purpose: To prompt the user to specify details about their selected course. This screen is necessary to allow our app to post the details about the course and fill out the course object

Interface Items:

1. Radio Button group that allows user to select between yes or no (for lab/no lab)
2. Text input fields to allow user to enter course name, location, start time, and end time
3. Checkboxes that allow the user to select/deselect which days their class meets
3. Submit button that confirms the information given by the user and adds that course to the course input screen



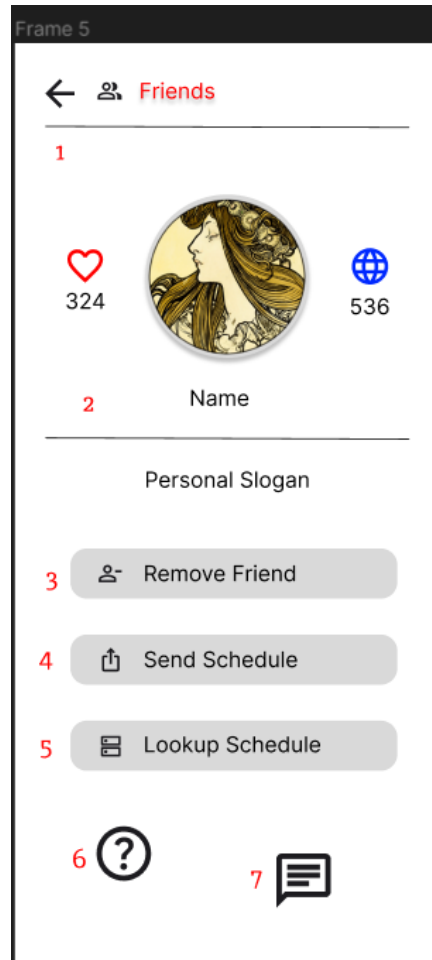
Name: Friends List Screen

Developed by: Robert Stickler

Purpose: To display the students friends on the application. The user can search for, add, and delete friends from this screen to enhance their network and compare life schedules.

Interface items:

1. This search button can be used by the student to search among current friends. The user can search and find their friends easily to be able to interact with them by looking at their schedule or messaging them.
2. The student can press the left button to add a friend and the right to remove someone that is their current friend.
3. The scrolling bar can give a student the ability to scroll through their friends in order to find someone that they can then click on in order to view that person's profile.



Name: Friends Profile

Developed by: Robert Stickler

Purpose: To display a friends information in a nice easy to see format with plenty of options

Interface items:

1. Back arrow to go back to friends list
2. Profile view with number of likes your profile has, number of friends, name, a slogan which can be like a small bio and finally a picture that you can customize.
3. A button to remove that friend from the friends list
4. Send your schedule so they can compare
5. Lookup their schedule to compare for yourself
6. A help button help understand what is happening on the screen
7. Be able to message that friend you are looking at.