Design Studio 1: RobotKid

Discussion 5 Team 2

Tong Ge

Richard Hsu

Kerrie Garcia

Payam Dowlat Yari

Noreen Angela

Goals

- User should have **freedom of control** over RobotKid.
- RobotKid should be **easy to use**.
- RobotKid **prevents** the primary user from **cheating**.
- RobotKid is **consistent with pre-existing video call applications**.
- RobotKid enable **effective communication** between teacher and student.

Constraints

- RobotKid have **limited battery life** of 7-8 hours max.
- RobotKid **requires stable internet connectivity** to operate.
- RobotKid **cannot open doors**, **go up or down stairs**, or **move in non-flat surfaces**.
- RobotKid have **limited Speed** and can at most go a bit faster than the normal human walking speed.
- RobotKid **cannot sense motion** around them like humans.

Assumptions

- Students are **physically** able to **operate** the software.
- Students have **access** to a **device** that can operate software.
- **Charging ports** will be **available** all around the school.
- RobotKid will have **basic motor functions** that the primary user can **control**.

Audiences

- (primary user) **K-6 students** that cannot go to school
    - Because of injury
    - Because of immune deficiency diseases
    - Because of family events, delayed flights, etc.
- **Elementary School teachers** who have to teach the primary user.
- **Peers** of primary user that will need to interact with RobotKid.
- **Parents** of primary user that may need to help their child operate RobotKid.
- **School faculty** in charge of charging, locating, distributing, updating, and fixing RobotKid.

Other Stakeholders

- **U.S. Department of Education** makes laws and regulations on what RobotKid can or cannot do.
- **School administration** has the power to set standards on what kind of students can use RobotKid, when students can use it, and how RobotKid can be used.

Main Functionalities

Video Call Software Functionalities

Starting out, this software will follow the conventions of most video calling software that are out there (Skype, Facebook Messenger, FaceTime... etc). Once the user connects to the robot of their choosing (see the Student's View in the Workflow section), the screen will change to a full-screen video feed from the robot's camera. The menu will appear as a hamburger menu button on the top right corner. When the menu is clicked, there will be multiple buttons will be displayed including:

- Hand-raising button
- Scratch Pad button
- Mute Microphone
- Log-Off

Figure 1. UI design for the student's view of the classroom.

Screen-Sharing

The screen-sharing feature is implemented for cases where teachers may want to show things from their PC onto the projector. Because Wi-Fi connection quality can vary, video quality may be spotty and so the user can't see the projection clearly. To bypass that, there will be a version of the web application for teachers to use in order to better communicate with the user of the robot.

After the teacher user have logged on to their account on their machine, the software will require them to enter a confirmation code. This code will show up on the RobotKid's display after the student has logged in.

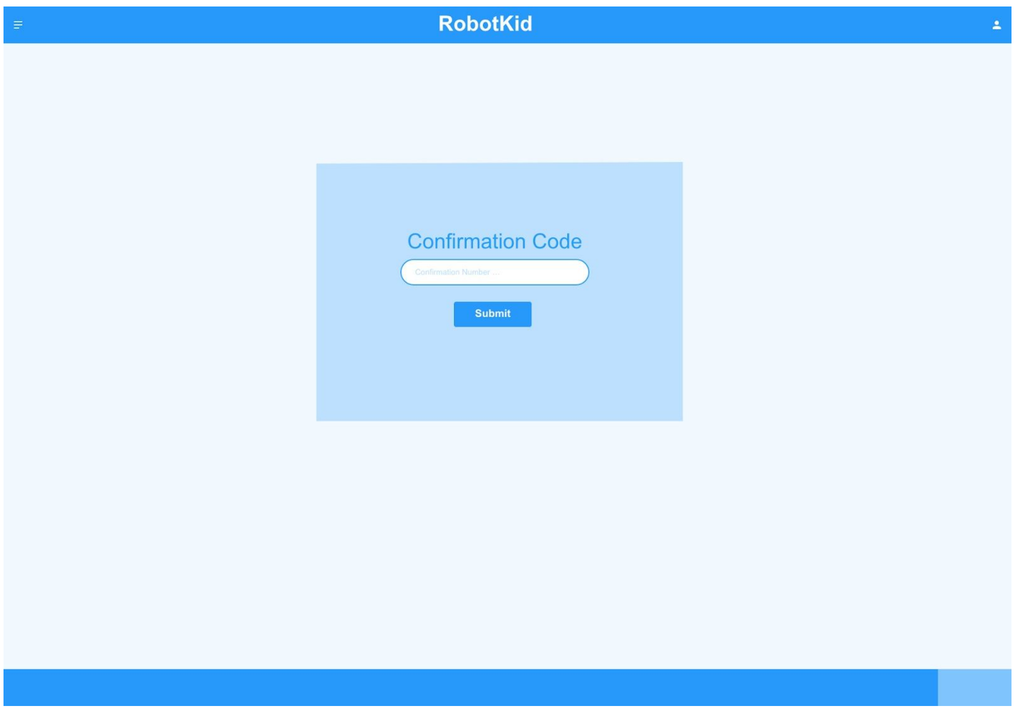


Figure 2. UI design for the teacher's view after logging in. The teacher inputs the confirmation code displayed on the RobotKid, so the system knows which RobotKid student to share the screen with.

The teacher user will type in the confirmation code that the student's tele-robot will show on screen. When that code is entered, the teacher has direct access to the student user through two modes:

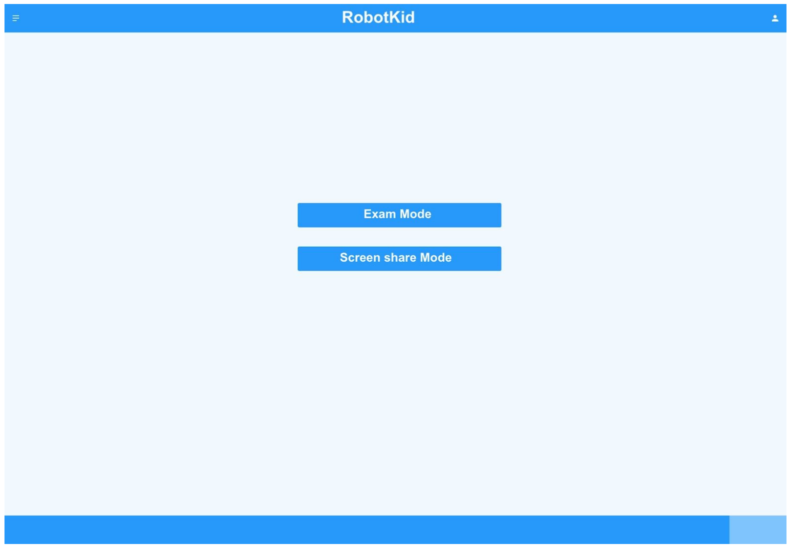


Figure 3. UI design for the teacher's view after confirmation code. Two modes are available to choose from.

1. Exam Mode
    a. This mode will allow the teacher to view the student user's face cam.
    b. This will serve the purpose of preventing the student user from attempting to cheat by pulling out their phone
2. Screen share Mode
    a. This mode will allow the teacher to share their PC screen with the student user.
        i. The screen from the student user's POV will first ask if the student would like to see the instructor's screen.
        ii. After the student user agrees to see the teacher user's screen, the view of this screen would result in a split-screen view of the classroom and the teacher's screen

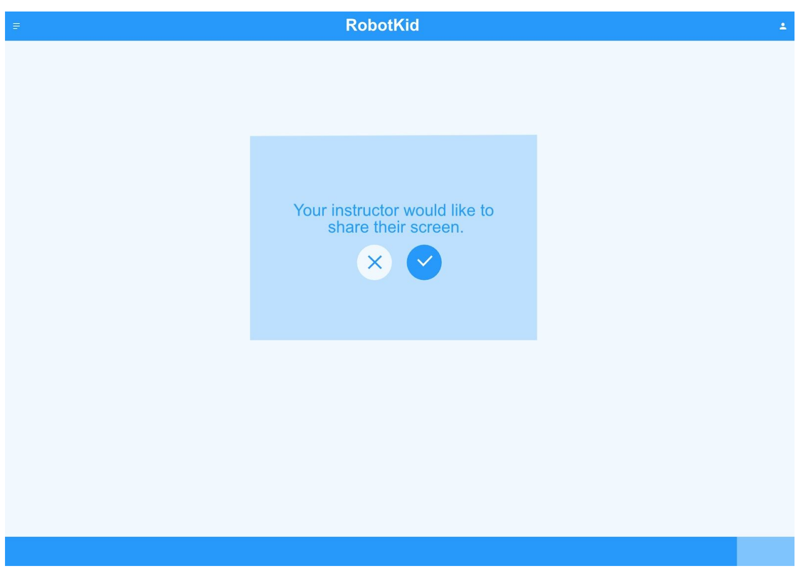


Figure 4. Confirmation for the student after the teacher selects "Screen share mode".

Free Control of RobotKid

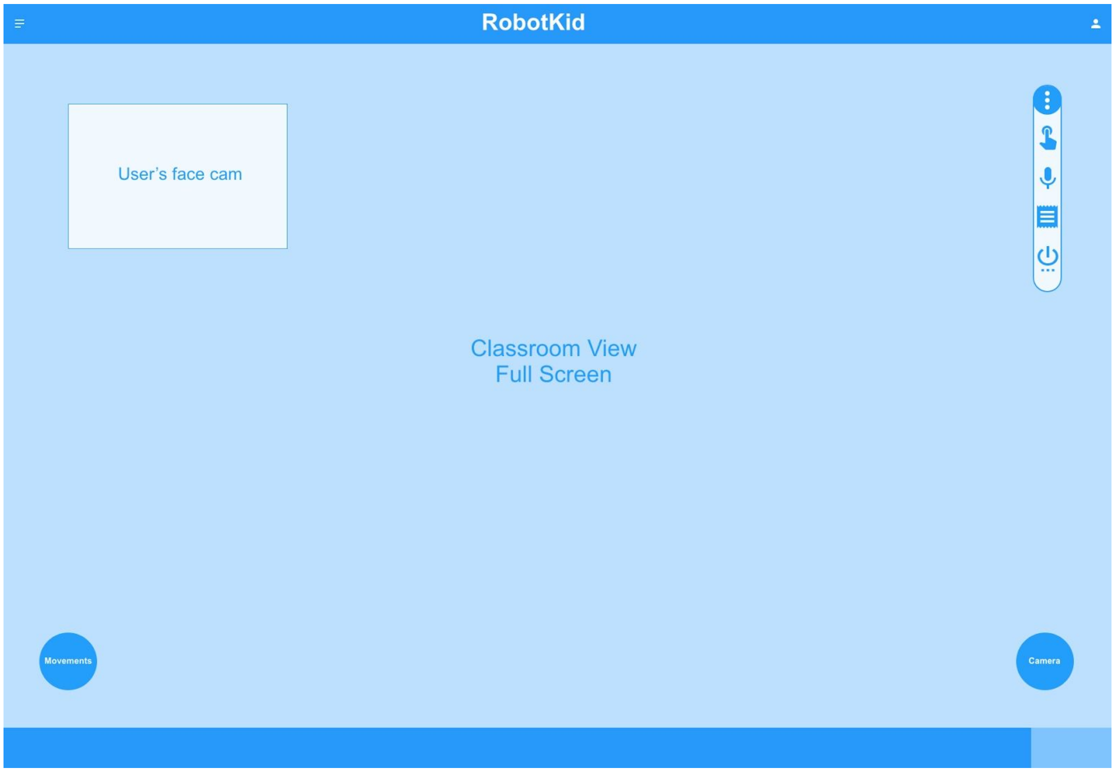


Figure 5. UI design for the student's view of the classroom. Two touch joysticks control are overlaid on top of the classroom video. One controls the movement of the RobotKid (left), the other controls the camera rotation (right).

One of the main decisions our team made for the KidRobot software was to give the main user full control of the tele-robot. How we are implementing this in the software depends on where the website is opened. If the device accessing the website is on tablet, then the software will provide two touchscreen joysticks positioned at opposite sides at the bottom of the screen. The left joystick will control the angling of the camera itself, and the right will control the movements of the actual robot. On the other hand, if the website is being accessed from a laptop, then controls will be delegated to the WASD keys for camera and arrow keys for movement.

Our main reason for making this design choice was to maximize the user's free-will while attending school through this tele-robot. For example, what if the teacher wants the students to choose their own groups for an activity? The ability to have freedom of where to go will help the user socially because they can move to whoever they want to engage. We intended this design to be put in use more during break times. During lunch, for example, the user can move their robot to the table with their friends and they can enjoy socializing that way.

Hand-Raising Button

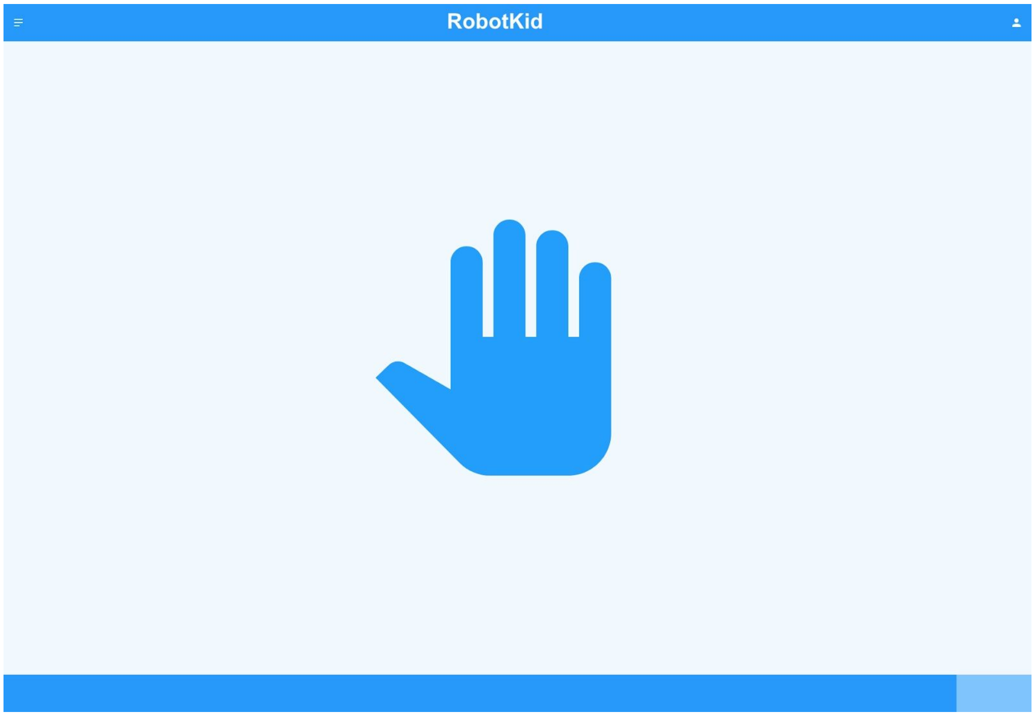


Figure 6. Display on RobotKid's screen to viewers on the other side when the hand-raising button is tapped.

Another main functionality included in our software is a button designated to notify the teacher when the user wants to participate. Since the tele-robots do not have any other appendages to move, communicating these actions that are such a staple of the student experience was something that our software needed to accomodate.

The button is placed in the side menu above the other buttons and is represented with a hand. When pressed, the button will invert colors to indicate it is pushed and then the telerobot's screen will switch to a clip of an animated waving hand. The state of the screen will stay that way until the button is pressed again, returning to the user's facecam. We decided against adding any additional auditory signals to notify the teacher because in a classroom setting any kind of repeated noise becomes a nuisance.

We considered having a multitude of "emojis" available as a menu for the user to communicate their emotions to others, but we decided that those kind of features were unnecessary for the problem we were trying to solve, so we kept just the hand-raising as a single button.

Answer ScratchPad

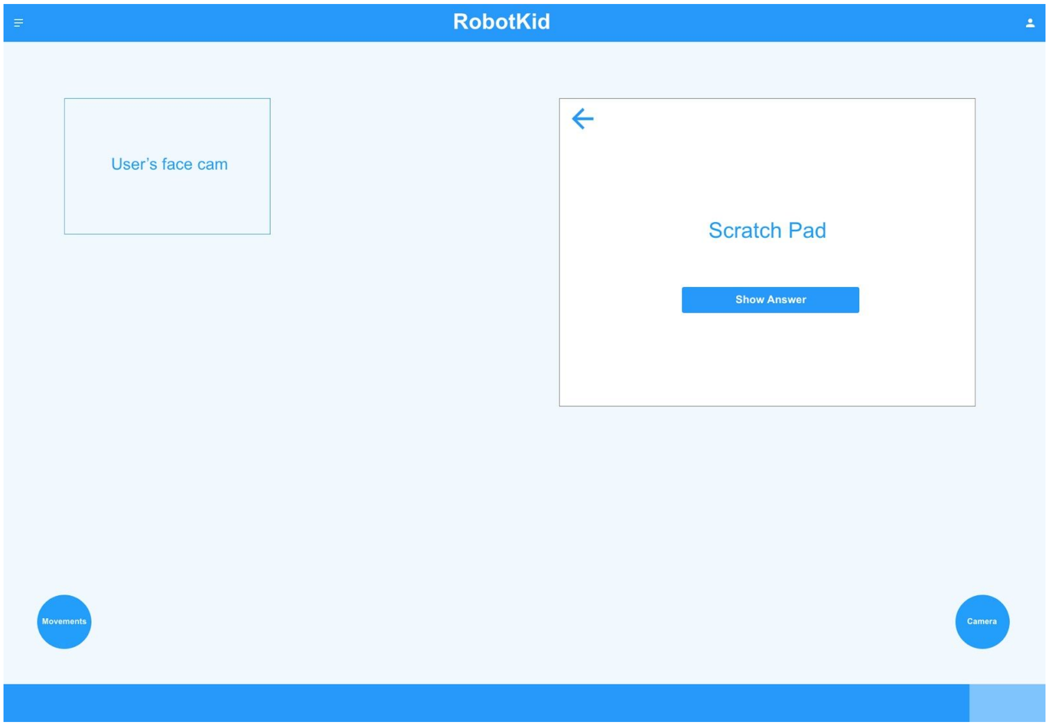


Figure 7. UI design for the student's view of the Answer ScratchPad. It is overlaid on the top right of the screen. An Answer button appears at the bottom center of the ScratchPad and a back button is available on the top left corner.

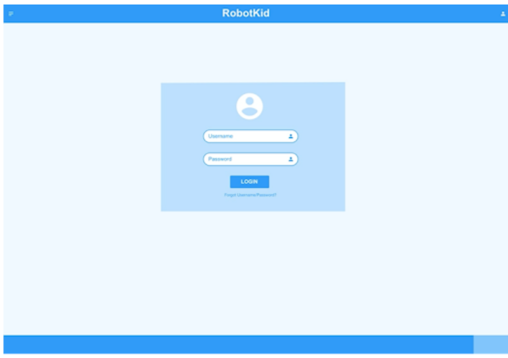
Under the hand-raising button, is the button that allows users to use the ScratchPad functionality. When the user presses this button, a small window appears in the corner that allows the user to write on it using either their mouse or their hand (depending on the hardware). We intend to implement this as a rudimentary drawing app with only a single pen. Since the main point is to communicate a small message, that part of the software does not have to be more complex. At the bottom of the window, there will be a button that says "Show Answer" and by pressing this button, the RobotKid screen will change to show whatever the user wrote. This function is intended to be another mode for the user to communicate with the teacher and others in cases where the user doesn't want to speak, or can't.

Tutorial

When the primary user create an account, they are required to complete a tutorial on how to smoothly navigate through the software. The tutorial would consist of a step by step interactive visual demonstration of each functionality of the software. Detailed instructions of each feature will be voiced over to its corresponding visual demonstration. Voice overs will be narrated in a slow, direct, child-friendly way with no difficult vocabulary or complex instruction. Interactions of visual demonstration will have arrow indication with key actions emphasized in a bright color. Having the tutorial enable the user to navigate through the software more easily.

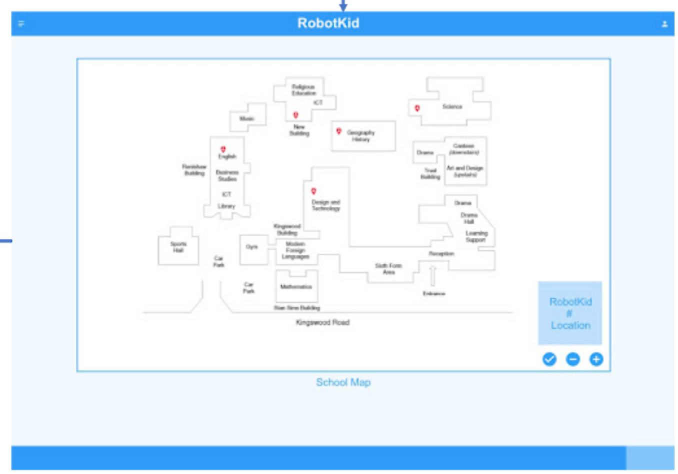
Workflow

Student view:



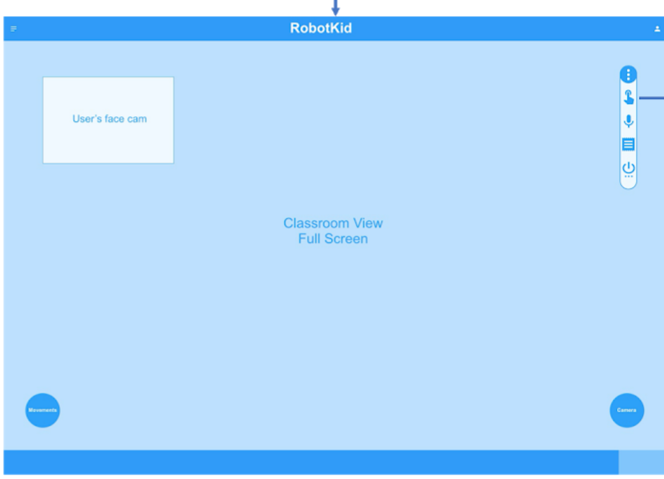
Initial log-in page for the student.

Student logs in with student ID



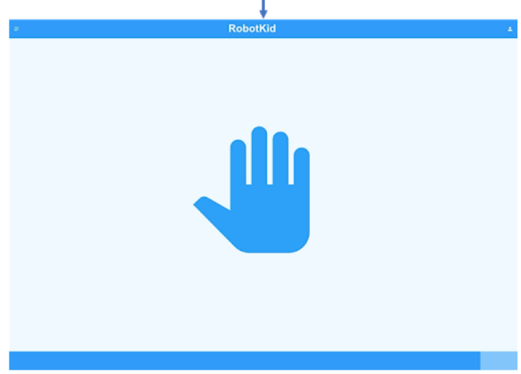
A map of the school is displayed with the locations of available RobotKids (shown by red markers).

User selects a RobotKid on the map



Student's view of video calling session. The classroom view covers the entire screen, and the student sees their own front camera on the top left corner. On the top right corner, there is a menu with "raise hand," "mute," "scratch pad," and "log out" buttons. Two touch-input joysticks are also available on the bottom corners. One moves the RobotKid (left) and one moves the camera (right).

User taps "hand-raise" button



UI shown on the RobotKid's display in the classroom. The hand-raise signal is semi-transparent and overlays the student's front camera video.

Two Alternative Approaches

Alternative #1 - Virtual Reality Design

Goals:

- RobotKid should be **easy to use**.
- Provide an **immersive classroom experience**.
- Allow students with low self-confidence to express themselves through **personalization**.

Constraints:

- Requires students to purchase a **phone VR headset**.
- Requires students to have access to a **mobile phone**.
- RobotKid is **limited to moving only within one classroom**.

Assumptions:

- It is possible to **mount a VR camera** on RobotKid.
- Students have **knowledge of how VR operates**.
- Students **do not get motion-sick while using VR**.
- RobotKid will have **basic motor functions** that the teacher can **control**.

Audience:

- **K-6 students** (the primary user) that cannot go to school **due to limited motor ability**.
- **Elementary school teachers** who have to teach the primary user.
- **Peers** of primary user that will need to interact with RobotKid.
- **Parents or nurses** of primary user that may need to help their child operate RobotKid.
- **School faculty** in charge of charging, locating, distributing, updating, and fixing RobotKid.

Main functionalities:

For this alternative, we focused our efforts on our new goal, which is to allow the student to have the most immersive, social, and personal experience. Based on these goals, we decided to implement a Virtual Reality system. We plan on implementing a small VR camera on top of the RobotKid to capture 360 surround video in real-time. We then send this video to the student connecting to the RobotKid.

For further ease of access and monetary considerations, we decided to implement our system on mobile for both VR-ready iOS and Android devices. One reason is because we assume that most modern students will have access to a smartphone (whether they own one themselves or their parents own one). Another reason we decided to go with a mobile app is because research showed that the current market price for cell phone VR headsets are cheap and ranges from $10 to $100. This factor was a considerable part of our decision, because this option provides parents who are already paying hefty medical bills for their students to have a cheap alternative to RobotKid.

Finally, in the student's cell phone VR headset, they will be able to see the 360 degree live video feed of the classroom. Along with the live video, the system also includes closed captions, which cater towards students who have difficulty hearing. Furthermore, while the VR headset is on, the RobotKid's facing direction will change according to where the student is looking. This allows for the classmates to interact face to face with the student, providing a more immersive social experience for the student.

To improve on the social experience even more, we decided to let students express themselves even in a hospital bed by implementing a RobotKid virtual persona that they could personalize. The virtual persona will then be displayed on the RobotKid's display instead of a video of the student in the VR headset. This could allow students who are hospitalized, and not confident in their body image to express their true personalities through the avatar. Students can customize their hairstyles and clothing every day before they log in.

When we started discussing this alternative, we wanted our audience to be students who have limited bodily functions (for example, no limb mobility). Based on this, we had difficulty implementing movement of the RobotKid into our design. There was no way for our audience to control the RobotKid through VR joysticks. However, we still felt that it is still important for the student to move and participate in classroom activities. Therefore, we decided to allow the teacher, in cases of classroom activities, to have movement control of the RobotKid.

Disadvantages:

There are two main reasons why we did not choose this alternative. One problem comes with the introduction of the avatar personalization. Although it does provide self-confidence, personalization abilities, and an immersive experience of the classroom environment for hospitalized students, his/her classmates will still feel a sense of separation from the student. Since the avatar will be displayed on the RobotKid instead of a facial video of the student, his/her classmates can only see the avatar and not the real student's facial expressions. Thus, there is still a disconnect in the social interactions between the student and classmates.

Another big issue that this design presents is a matter of free will on the part of the student using RobotKid. The audience of the design targets students who have disabilities resulting in limited limb movements. Because of this, it is difficult for the students using RobotKid to provide any touch controls inputs to our mobile system. Initially, we tried designing a system using voice control rather than touch inputs. However, we quickly realized some issues with voice control:

1) The student's voice control instructions will disrupt a quiet classroom environment.
2) It will still be hard for the student to control the RobotKid's movement with voice-control. For example, he/she will have to constantly specify which direction to move towards, and the distance.

These limitations ultimately led us to our current solution of having the teacher control the movements of the RobotKid. Although this decision allowed the RobotKid to move, it also sacrificed the student's freedom. As our essence and goal was to create the best learning environment, our team firmly believed that the free will of the student should play a significant part. This limitation on freedom thus led us to choose our final design (which has freedom of control over the RobotKid through joysticks) over this design.

Alternative #2 - Interactive Design

Goals:

- RobotKid should **increase interaction** between user and peers
- Help users **make and maintain friendships**

Constraints:

- RobotKid have **limited battery life** of 7-8 hours max.
- RobotKid **requires stable internet connectivity** to operate.
- RobotKid is **limited to moving only within one classroom**.

Assumptions:

- User and school will **have an adequate wifi connection**
- User's device will have an **ample amount of storage space**
- RobotKid software can **interface with other software**

Audience:

- **K-6 students** that cannot go to school
- **Peers** of primary user that will need to interact with RobotKid.
- **Elementary School teachers** who have to teach the primary user.
- **Parents** of primary user that may need to help their child operate RobotKid.
- **School faculty** in charge of charging, locating, distributing, updating, and fixing RobotKid.

Main functionalities:

For this alternative, our main goal was to increase the interaction between the user and their peers. Our goal would help students make and maintain friendships despite not being able to physically attend school. With this goal in mind, we decided on implementing a system that focuses more on fun interactions for the user rather than the academic learning.

Our system would be implemented as a software application available for users to download. One reason for this is because we assume that the system would require a lot of storage space and multiple updates to maintain a fun experience for users. We also assume that the user and the school will have good wifi in order to accommodate the main features of this system. Our last assumption is that the system can interface with other software.

To increase interaction between the RobotKid user and their classroom peers, we decided to implement interactive minigames. The user would be able to play against or with their peers in real time with the user using their device and the peers using the RobotKid tablet interface. These minigames would allow the students to interact with each other in a fun, competitive or cooperative way.

Our system would also allow the user's peers to Airdrop to the user's device. The classmates would be able to send the user memes or pictures that they would normally show a friend in person on their phone. This feature will allow the user to feel closer to their fellow classmates and feel included thus improving interaction between the user and their peers.

Finally our system would implement a sketchpad and/or notepad feature. The user would draw on their device and what they draw would appear on the RobotKid's screen for the user's classmates to see. The classmates would then be able to interact with the feature by letting them draw on the RobotKid's screen in real time with the user. This feature would have personalized brushed and pens as well as an array of colors to choose from to foster creativity among the user and their classmates. We decided this feature would be good as it allows the user and classmates to interact more in a more creative way that otherwise would not be available if the user was not physically attending class.

Disadvantages:

Although this alternative helps simulate the social experience of school, we decided not to go with this idea. One reason we decided against this idea is the likelihood of a much longer development time. This system would require much more complicated features and planning than what we decided for our final version of the RobotKid. This would result in a higher cost of the overall project. This version of the RobotKid could also have a higher potential of disrupting the classroom environment. The user's classmates could end up getting distracted by all the fun, nonacademic features of this version which would result in the classmates not paying attention in class. Lastly we agreed that with the features included in this version of the robot, the school board may not approve of the use of RobotKid in their schools. This would be bad for the company as well as the potential users who are not able to attend school.