Magpie

Spring 2017

1. Introduction

1.1 Purpose and Background

The Rapid Design and Prototyping for Computer Systems class of Spring 2017 was given the challenge by the Children's school at CMU to help the school track location and activity of its students. The visionary scenario outlined in Phase 1 not only overcame that challenge, but modernized the inefficient process of dismissal at present and improved communication between teachers and parents.

Phase 2 saw the class work within their subsystem teams to design a solution to the visionary scenario presented in Phase 1. The team and product name of *Magpie* was chosen, after a friendly bird that has a startup-esque sound to it. Within Phase 3, each of the subsystems of *Magpie* was implemented and tested by various degrees. We hope that this project will be continued and fully implemented into the workflow and infrastructure of the Children's School.

1.2 Group Members

To help realize the visionary scenario and design goals, the class decided to divide its members into five teams. Below they are listed, along with their members.

Frontend Team: to develop dismissal assistance display and dismissal messaging display
Members: Leann Bahi, Jack Gao, Abhishek Jindal, Sarah Klein, Sujay Kotwal, Kavya Kumar, Luke Miller

Data Analytics Team: to analyze data sent from the dismissal team and to provide meaningful input for the frontend team
Members: Radhika Amare, Kevin Chon, Karim Elmaaroufi, Vanshaj Sikri, Sam Xu, Sheryl Zhang, Abhishek Yadav

Activity Sensing Team: to track students as they engage in various school activities
Members: Paola Aguilar, Aditi Chalisgaonkar, Kevin DeLand, Matt Harding, Dillon Lareau, Anand Prasanna, Vishnu Razdan, Neil Ryan

Dismissal Team: to expedite the task of student dismissal and make it more efficient
Members: Arani Basu, Sujay Kotwal, Kyuin Lee, Sally Lee, Jennifer Meagan Lloyd, Amanda Marano, Kyle O'Rourke, Akshit Sharma

Backend Team: to provide support for the activity sensing team, dismissal team, and frontend team by creating a common hub for data storage
Members: Momina Haider, Abhishek Jindal, Gayatri Kamat, Karen Zhou

2. Conceptual Design

The conceptual design section elaborates on the baseline scenario, in which a typical day at the children's school is presented. It highlights the problems, and thus the needs of certain technologies, the designs of which are discussed by the subsystems that are responsible for them, respectively.

2.1 Problem Definition

The Children's School at CMU is seeking to manage the research, kids' activity tracking, and dismissal systems more efficiently through means of technology. Currently, there's not a comprehensive system that takes care of teacher/parent communication, which is essential in determining if a child can participate in research, activity tracking, which is of high interest to both the parents and teachers, and dismissal, which could become a source of frustration for both the parents and the teachers if it's not a smooth process.

The below section describes a typical scenario in the Children's School. The baseline scenario accounts for the difficulties faced by children, teachers, and parents, in a typical day at the school, and provides all the subsystems a sense of the needs of the school.

2.1.1 Baseline Scenario

Mrs. Smith is a kindergarten teacher at Margaret Morrison Research School. She reaches school everyday at 8 am to prepare for the activities she will be conducting for the students each day. Today she is planning to take up an interactive session on the virtual board, which will involve a lot jumping around and enthusiastic participation.

Mrs. Jones is Ally's mother and she drops her to school every morning at 8.30am. As the students arrive Mrs. Darsh, waits for all the students to arrive to know if everybody will be there for the activity. After the students settle in, she finds a note inside Ally's bag. It mentions that she has not had a good sleep last night and might be cranky throughout the day. Mrs.Darsh is disappointed and wishes that if there was some way she could have known it previously, and then she would have made some other arrangements for her.

After the activity time, there is choice time for children where they can pick up an activity of their choice to carry out. As the children disperse into different activities, Mrs. Darsh has to take a mental note of who is doing what because she has to send an email to the parents about the activities held at the school.

In the meantime Aron, who is a researcher at the school, takes Ally away for a 'Research Game'.

Aron almost forgets to fill out the research tracking sheet, which is used to keep track of compliance issues. He adds Ally's research number to the sheet and then begins the study, which is playing a game for 20 minutes. Ally isn't in a great mood and Aron worries the results may not be useful for the study. He returns to the office, checks her back in, and Ally continues with her day.

At 12.30 Ally has to wait for Mrs. Jone to pick her up. She is already exhausted and can't wait to go back home. Mrs. Jones is running late because she is behind in the car line, which forms during pick up everyday. She is sulking because she is getting late for an important meeting in the office. Mrs. Darsh has to constantly look through the binoculars to find out which child has to be sent downstairs to be picked up. As Ally is in a bad mood she is throwing a tantrum now. Mrs.Darsh is having a tough time managing the pick up procedure as well as the impatient children.

After about half an hour of chaotic dismissal process, Ally finally gets picked up. Mrs. Jones finds out that Ally was taken for a research inspite of not being too well. She wants to talk to Mrs. Darsh about this but can't as she already running late. After the dismissal, Mrs. Darsh has to clean up and subsequently sends an email to all the parents about the activities held in the school. She is already tired and has no energy left to carry out these tasks.

2.1.2 Key System Requirements

Our research led us to focus on three core functions: student dismissal, student activity tracking, and communication. Each of these functions will be supported by multiple technologies and subgroups.

2.1.2.1 Dismissal

The Children's School dismissal process certainly jumped out during our research. Constrained by a narrow road that dead ends, dismissal at the Children's School is a chaotic process that involves binoculars, paper signs, and three-point turns.

To improve this process, the Children's School faculty must be able to know which student's parents are upcoming in the dismissal line. They also need to know when students have been dismissed, even if they've been dismissed by alternative means (for example, a parent parks their car and walks into the school).

The parents' key requirement is spending less time waiting in a dismissal line. If they're required to bring identification, it should be easy for them to remember it and they should have access to multiple copies in case multiple family members or guardians pickup their child.

2.1.2.1 Student Activity Tracking

The Children's School faculty and parents pride themselves on doing things the right way. As a research school, they are comfortable using data-driven approaches to improving the educational experience. We sought a solution that would augment their approach, by analyzing and storing each student's behavior at a low granularity.

For teachers, they must be able to access distilled information that gives them a clear picture of how students interact with each other and their activities. They then need the ability to put this information into action, by planning and storing new activities. Parents need the ability to relate to their child, by knowing which activities their child participated in each day.

2.1.2.1 Communication

Communication is clearly an important part of the Children's School experience. With that in mind, we were surprised to find how much important information was shared through paper notes packed into a child's backpack.

Parents need a centralized place where they can access important Children's School information, like newsletters, calendars, and announcements. They also need an easy way to communicate important information to the Children's School staff.

Teachers need a more sophisticated way of collecting and storing notes from parents. Multiple teachers need access to each note, along with a record of when the note has been addressed by someone on staff. Given the restrictions on phone use in the classroom, teachers will need a tool to access to notes in the classroom, which balances accessibility with privacy.

Category	Item	Requirement
Dismissal	Display Interface	The interface is easily accessed by appropriate staff. The interface displays real-time data relayed from the dismissal sensors. Interface allows teachers the flexibility to handle corner cases, where parents pick up their child in different ways.
Communication	Parent interfaces	Easily used from mobile device. Consolidates all of the information they regularly need. The ability to share notes based on category.
	Teacher interfaces	Parent notes are easily accessed by multiple teachers. History of notes are kept on record. Notes have clear categorization.
Student Activity	Teacher interface	Provides clear information on

Tracking		activity interaction and student behavior. Teacher can easily adjust and plan lessons. Teachers can review historical data and past activities.
	Parent interface	Automatically shares their child's daily activity.

2.2 Initial Solution Concepts

Upon learning about the current situation as demonstrated in the baseline scenario and learning about the functional requirements, we began to conceptualize solutions to create a system that connects data sensing, processing, storage, delivery, and display, utilizing standard sensors, common database solutions and frontend display. With these technologies in mind, we began to frame use cases of our system, and developed a visionary scenario.

Table 2.2.1 Table of Selected Technologies

Subsystem	Selected Technologies	Functionality
Backend	PostgreSQL	Database management
	Express	Node js web application framework
	Local Machine	Local server that runs database
Frontend	Balsamiq	Low fidelity wireframing tool
	Figma	Collaborative high fidelity wireframing tool
	Javascript	Web development
	CSS & HTML	Web development
Data Analytics	Python	Processing data
Activity Sensing	Chafon UHF passive RFID tags	Tags will be placed in wearables
	Chafon 6M Medium-Range Readers	Readers will communicate with tags to track students
	Raspberry Pi	Used to pre-process data and send data to local database
Dismissal	RF200 active RFID modules	Used as both receiver and as tags given to the parents.
	Raspberry Pi 3	To take data from the reader and transmit to the backend server over WiFi.
	Chafon UHF passive RFID tags	Alternate solution for tags.
	Chafon 6M medium range	Alternate solution for tag readers

	reader	

2.2.2 Visionary Scenario

Let’s look through a day in the life of Ally, Mrs. Jones, and Mrs. Darsh. Mrs. Darsh is a kindergarten teacher at Margaret Morrison Research School. She reaches school everyday at 8 am to prepare for the activities she will be conducting for the students each day. Today she is planning to take up an interactive session on the virtual board, which will involve a lot jumping around and enthusiastic participation.

Mrs. Jones is Ally’s mother and she drops her to school every morning at 8:30am. Before leaving the house Mrs. Jones notifies Ally’s teacher, Mrs. Darsh, through the school wide app (also viewable as a website) that Ally has not had a good night sleep and might be cranky throughout the day. She then makes sure that Ally has put on her wearable that has been provided by the school. In the meantime, Mrs. Darsh checks the school wide app for any updates or notes from parents. She sees Mrs. Jones’ note and plans accordingly to accommodate to Ally’s lack of sleep.

As the students arrive at school, Mrs. Darsh and her colleague, Mrs. Smith wait for them. As Ally and her mother drive up, Ally’s wearable communicates to the detection system to record her attendance in the app and the school’s unified database. Mrs. Smith takes Ally inside while Mrs. Darsh has a word with Ally’s mother letting her know she’ll look out for Ally throughout the day.

To begin with, the students have choice time, which allows them to play and learn as they please. Their wearables let teachers know where the students are at any given point of time, and also let teachers and parents have a look later on at how their children are spending their free time. It comes in handy when Ally’s friend, Francis, wanders into an area that is off limits for him. The staff gets notified, and Francis is redirected to other activities.

The school also has structured group time, in which all kids of a certain age come together. The themes and activities have been planned out by the teachers and logged into the school’s app/website. This allows Mrs. Darsh to pay full attention to the students - she doesn’t need to keep track of things for her afternoon email to parents because she knows that the app and wearable will do that for her. The students have a great time, though Ally is a little more cranky than usual.

In the meanwhile, Aron, a researcher at the school, arrives, sets up his experiment, and checks his app to see which children are available for research. These children’s wearables also visually indicate their availability. Aron is able to easily spot that Charlotte has already been taken in for research and is not available for research but Ally is. Aron asks Ally if she will play his game and she agrees. Through Ally's wearable and Aron's app, it is logged into the database that Ally is participating in Aron's research. The app lets Aron know when his 20 minutes are up and Ally returns to play with Charlotte. Ally's parents will later be able to view the information about the research through the app.

At 12:30 the students get ready for pick-up. As the cars line up, the parent's app sends their location to the database. The order of cars lined up is therefore logged and viewable by the teachers. Mrs. Jones arrives and she is added to the queue. Ally's wearable changes color indicating it is her turn to be picked up. Mrs. Smith takes Ally downstairs and to Mrs. Jones' car.

After the dismissal is completed, teachers get a bit of a breather before they have to finish their final task for the session - an email updating the parents on what their children have done during the day. Most of the information is already sent to Mrs. Darsh's version of the app. She looks it over quickly and sends it out to parents, who can view it on the app as well.

2.3 Conceptual Design

The following subsections describe the internal design that each individual group comes up with, which describes how they envision their subsystem's end-to-end functioning.

2.3.1 Selection criteria based on visionary scenario

Based on the visionary scenario presented in the report for Phase 1, features were segregated into different levels of priority. Priority 1 features were the "must have" features, Priority 2 were the "should have" features and Priority 3 were the "could have" features. For phase 2, all teams implemented the priority 1 features in order to have a basic prototype working and some teams even implemented the Phase 2 features. The selection criteria were based on having a minimum viable product working and then adding extra features to enhance its functionality later.

2.3.2 Product Design Specifications

The core implementation for activity sensing and easing out dismissal process was implemented using Raspberry Pi, passive RFID system and a 6-meter medium range RFID reader. The general reputation of Raspberry Pi, low power consumption of passive RFID subsystem, and good accuracy of RF were key in finilizing the product specification.

The specifications of Raspberry Pi 3 is given in the following table:

Raspberry Pi 3
1.2 GHz 64-bit quadcore ARMv8 CPU
802.11n Wireless LAN
Bluetooth 4.1
BLE
1GB RAM
4 USB ports
40 GPIO pins
Full HDMI port
Ethernet port
3.5 mm audio jack
Camera Interface
Display Interface
MicroSD card slot
VideoCore IV 3D graphics core

2.3.3 Front End

The frontend team is responsible for designing and building the interfaces that each user will interact with. At the highest level, our designs make it easier for Children's School parents and teachers get the information they need, when they need it. Specifically, we have designed solutions that improve communication, dismissal, and activity planning.

Each of these processes center on presenting data in compelling and useful ways. For parents, our focus is providing a simple way for parents to get and communicate information about their child, for example, a list of their child's activity for the day. For teachers, Magpie will help them better understand their students and how they interact with their surrounding environment. Examples of these visualizations can be found in our teacher dashboard designs. The dismissal experience is designed to reduce the stress and chaos that currently characterizes the Children's School dismissal process. This will be a simple display that will end the need for large signs and binoculars that are key elements of the current dismissal process.

2.3.4 Data Analytics

The Data Analytics group receives raw data from the frontend group as .csv files. We then scrape these files to extract the information and store it in sets. Post copying the data, the analytical functions are run on it and new statistics are collected. In Phase 3, we will implement the design to send this processed data back to the frontend team for visualization.

2.3.5 Activity Sensing

The Activity Sensing subgroup will utilize a passive RFID system to collect data regarding the students’ activities, such as the time spent at each activity station and presence of other students at that station, to allow individualized analysis of their learning tendencies and social network. Specifically, the passive RFID tags will continuously report their whereabouts and signal strength to base stations that will be placed in the ceiling above the activity stations. This rich data will be pre-processed before being pushed to a local database where it can be analyzed and output to displays. Finally, the system has been designed to be automated and non-intrusive in order to limit the disruptions in a student’s educational day.

2.3.6 Dismissal

The Dismissal subsystem is consisted of three processes: parent with RFID tag is registered by the RFID tag reader; the tagged data is processed; and then the tagged data is sent to the main server so that the faculty can determine which child is ready to be picked up. Our design implements RFID tags that send a string text that can identify the child that needs to be picked up to the reader. Once the string text is read, it is processed into a HTTP POST request and sent to the backend system via Raspberry Pi, which will then display the information to the faculty. Primary option is using active RFID tags with cigarette lighter power adapter and secondary option is to use the passive RFID tags with stock tag reader.

2.3.7 Back End

The back end team is responsible for setting up a central server and a database that act as the backbone for the whole system. The server is set up on a local Linux machine, and is responsible for supporting the database, as well as being a platform of messaging web application interface, children activity display, and vehicle tracking dashboard. The database, as the name suggests, is designed to store data about the children’s activities, the proximity of parent vehicles to the campus during dismissal time, and the processed data that’s used for data visualization. Both the raw and processed data is constantly pushed into the database in case of a need for data analysis over an arbitrary requested period of time in the future.

3. System Tutorial and Usage Scenario

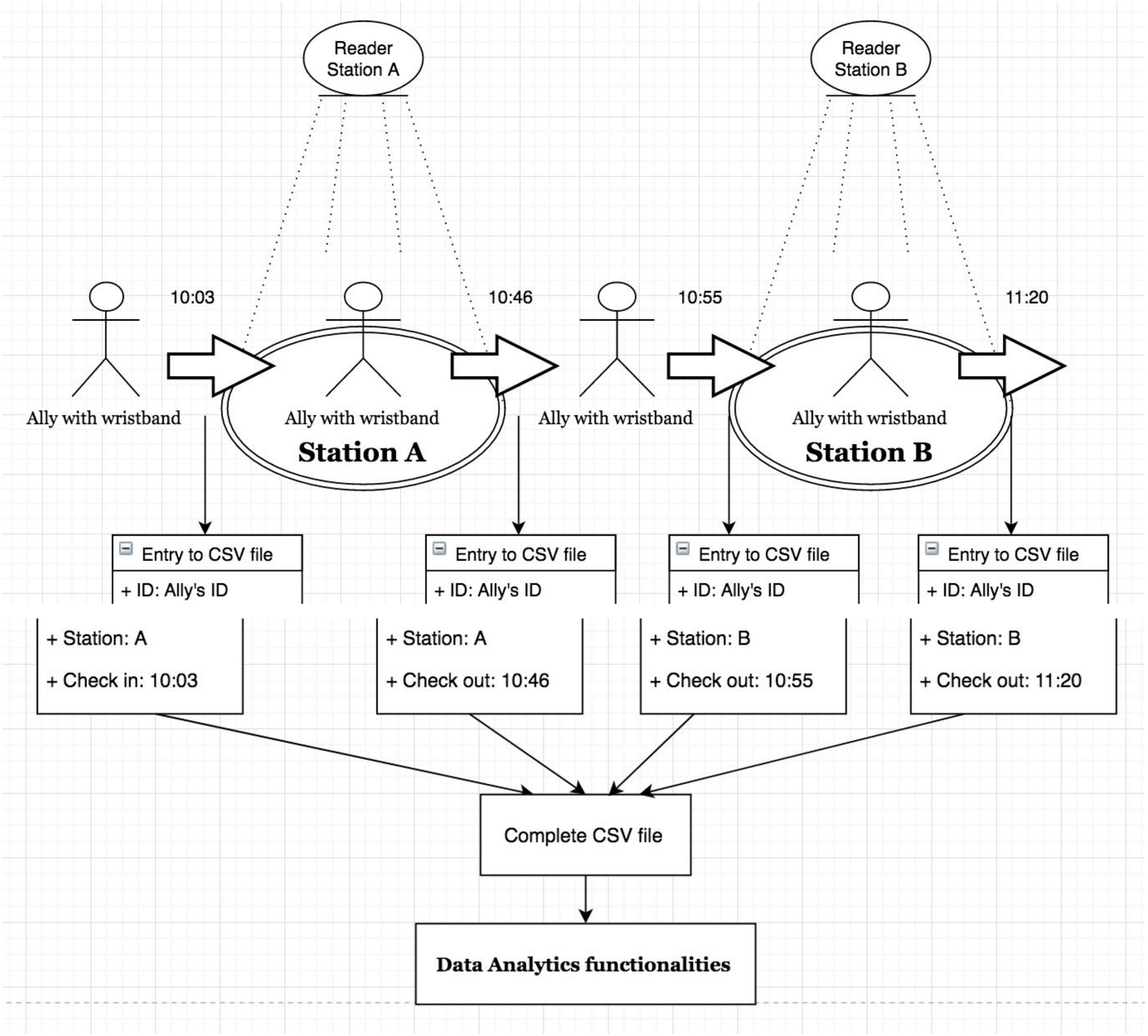
3.1 Front End Usage Scenario

Our in-group interactions focused on defining and delegating the tasks that needed to be completed. We individually created user stories and then collectively identified the roles and functionalities that we needed to design for, along with a few additional internal tasks.

Externally, we worked closely with the backend team. We also communicated with the analytics and sensor teams. The front end team will continue to work closely with these teams as the implementation phase ramps up.

3.2 Data Analytics Usage Scenario

This figure shows the usage scenario from the perspective of the Data Analytics subsystem



3.1 Activity Sensing Usage Scenario

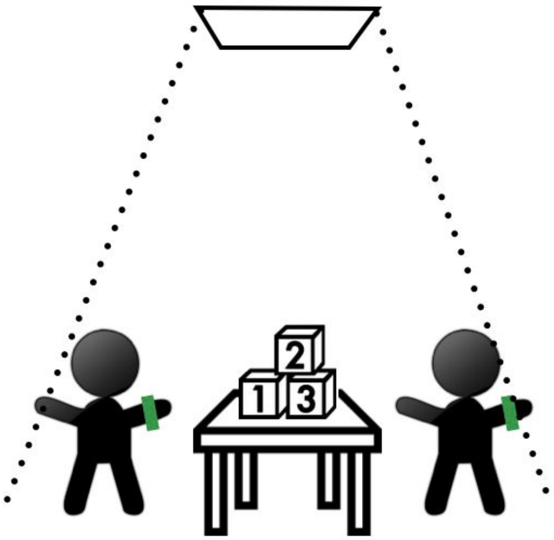
At the beginning of the day’s session: The children are expected to leave their wristbands at the school each day, and so they would need to put them on during the first routine in the morning. It would be optimal if an opportunity could be found within the existing morning routine for the children to perform this action. Accordingly, the team examined the morning routine:

- Take off things in hall
- Wash hands
- The day starts and ends at the rug
- Pick up a slip of paper from box, write their own name, and answer the question of day
- Put their names on the chart

The team felt that the wristbands could be worn as a part of this morning activity, at the same time as children put their names on the chart. This would ensure that it is after hands have been washed (to reduce chances of water damage), and it is an action which each child performs everyday at the beginning of each day.

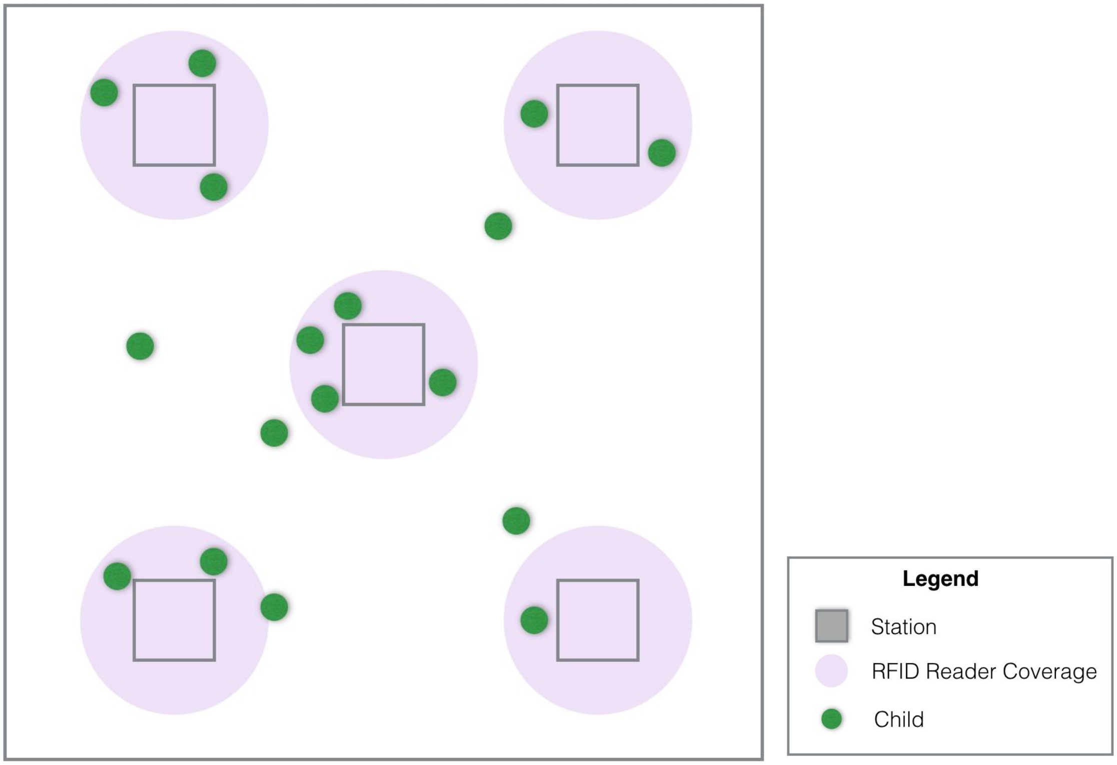
It also ensures that almost no meaningful information is lost. The wristbands are helpful in understanding patterns that emerge from the children's’ behaviour when it is less structure, and the morning routine is very well-known and structured. However, the activity following this is choice time, during which the team could extract meaningful patterns of data which the teachers themselves may not be aware of

During the day’s session: Over the course of the day, the best possible scenario is if the children were able to go about their day as if the wristband was not present. Thus, the team endeavoured to make the wristband as comfortable and unobtrusive as possible. Though there are options for sounds to be emitted when a tag (on the wristband) is sensed by a reader, the team has ensured that this will not be emitted as it would be intrusive. Thus, to the children, it should seem like nothing new, different, or special is happening.



Visualization of one station

On the data collection end, it is expected that data will be collected when the a child is in the range of a certain station (since each station will correspond to a single reader)



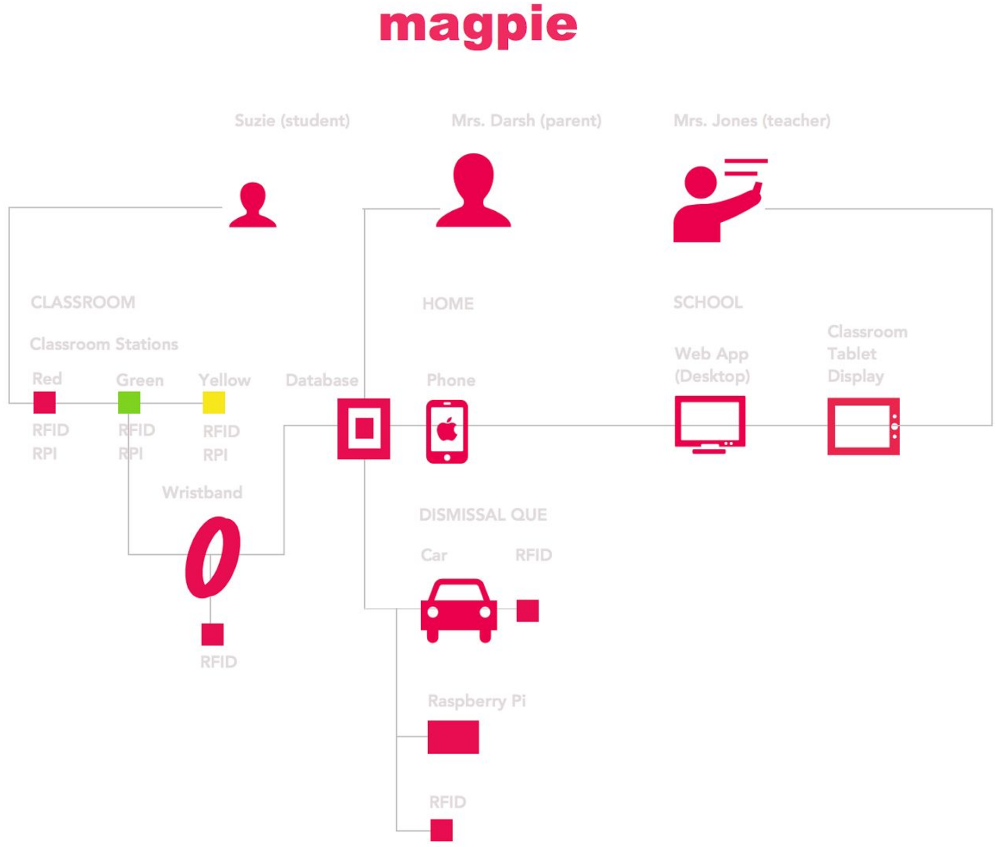
Visualization of floor

At the end of a day's session: As with the routine at the beginning of a session, the team searched for opportunities to for children to leave behind their wristbands at the school in a way that fit with their existing routines. One promising time and space is that reserved for the childrens' mailboxes - each child has a an individual mailbox. Since the wristbands do not need to be charged (being passive tags), each child can simply leave their wristband in their own mailbox, and pick it up the next morning

4. Implementation

4.1 System Overview

4.1.1 System Diagram with software and hardware modules



A diagram showing the interactions between different modules.

4.1.2 Technology Used in each module

Front End

The front end web system was served using the Node.js Express Framework.

Data Analytics

The data analytics software was written in Python. They use matplotlib and plotly to generate graphs for data visualization for the front-end team

Activity Sensing

The activity sensing team uses a Chafon RFID Reader to sense passive RIFD Tag wristbands. The information is processed using a Raspberry Pi.

Dismissal

The dismissal team uses a Chafon RFID Reader to sense Omni-ID Dura 1500 RFID Tags, and the information is processed using a Raspberry Pi.

4.2 Front End

4.2.1 Functionality

The frontend team identified three key end users: teachers, parents, and administrators. We designed unique interfaces and functionality of reach of these users. These screens are described in more detail in section 4.2.3 of this report.

Teachers
Magpie gives teachers the information and tools they need to make informed decisions in the classroom. The core of the teacher functionality is a desktop experience, which allows classroom activity planning and student data visualization.

In addition, the frontend team designed a dismissal assistance display and a digital messaging display. These interfaces will be used ad hoc by several teachers and will benefit from the accessibility provided by displays.

Parents

Magpie’s parent view allows parents to message teachers, read announcements, find links to key resources, and review their child’s top activities for the day. These interfaces are primarily designed for simplicity and mobility. Parents are already stretched thin and are regularly bombarded with a great deal of information. Magpie’s core value is in consolidating key information into one, easy-to-use experience.

Administrators

The administrator views focus on high-level organizational tasks. Functionality includes adding and removing users, assigning groups, and distributing announcements. As we found in our research, reliability and flexibility are important for any process at the Children’s School. With that in mind, the administrative capabilities make it easier to manage and adjust accounts as necessary. In this implementation and system integration phase, we successfully integrated our frontend interfaces with backend, activity sensing, and dismissal.

4.2.2 Interactions

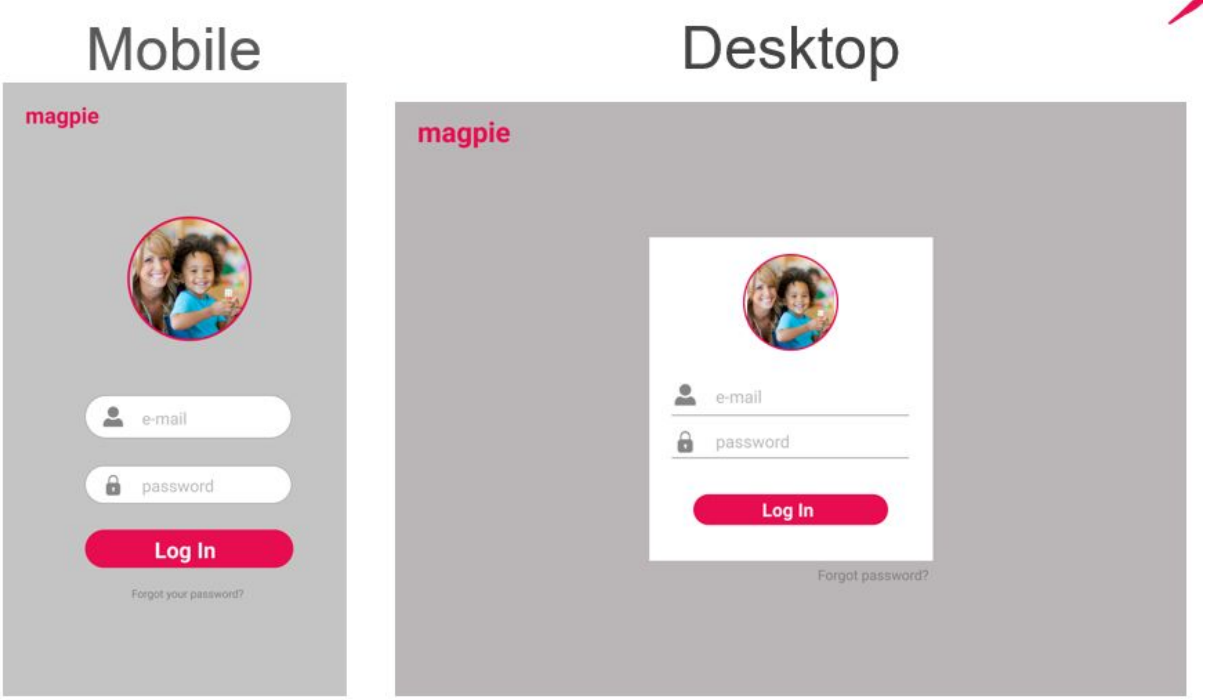
Our in-group interactions focused on defining and delegating the tasks that needed to be completed. We individually created user stories and then collectively identified the roles and functionalities that we needed to design for, along with a few additional internal tasks. Externally, we worked closely with the backend team. We also communicated with the analytics and sensor teams.

4.2.3 Screens

Our screens designs are below. You will see a combination of medium fidelity wireframes, as well as examples of what the final implementation may look like.

Login

Magpie will have a simple login screen. This screen will include a to-be-determined logo and the ability to reset a password if needed.



Mobile

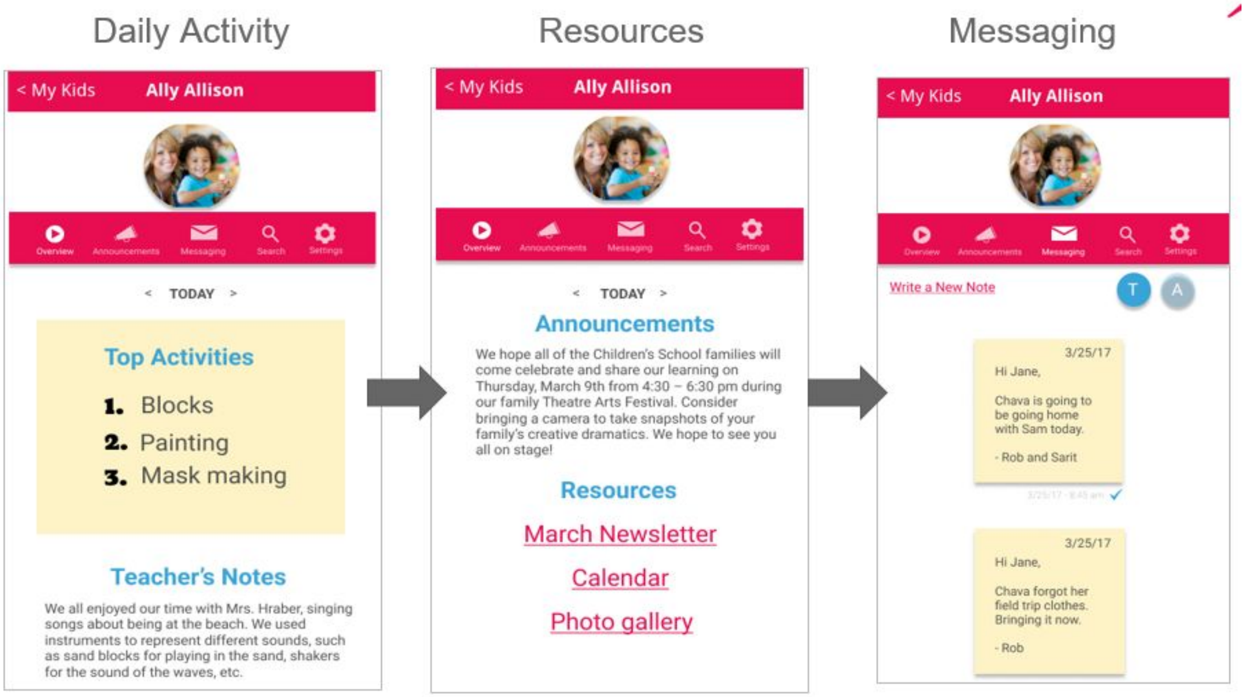
Desktop

Parent Screens

As described above, the parent screens focus on mobility, ease-of-use, and communication.

After logging in, the parent will be prompted to pick from a list of children. This page design presents the image of the child as a central part of the experience, while also solving for parents that have multiple children at the Children's School.

After selecting a child, the parent will be taken to the home screen. From here, they can quickly find information using the navigation bar.



Top Activities

1. Blocks
2. Painting
3. Mask making

Teacher's Notes

We all enjoyed our time with Mrs. Hraber, singing songs about being at the beach. We used instruments to represent different sounds, such as sand blocks for playing in the sand, shakers for the sound of the waves, etc.

Announcements

We hope all of the Children's School families will come celebrate and share our learning on Thursday, March 9th from 4:30 – 6:30 pm during our family Theatre Arts Festival. Consider bringing a camera to take snapshots of your family's creative dramatics. We hope to see you all on stage!

Resources

March Newsletter

Calendar

Photo gallery

3/25/17

Hi Jane,

Chava is going to be going home with Sam today.

- Rob and Sarit

3/25/17 · 8:45 am ✓

3/25/17

Hi Jane,

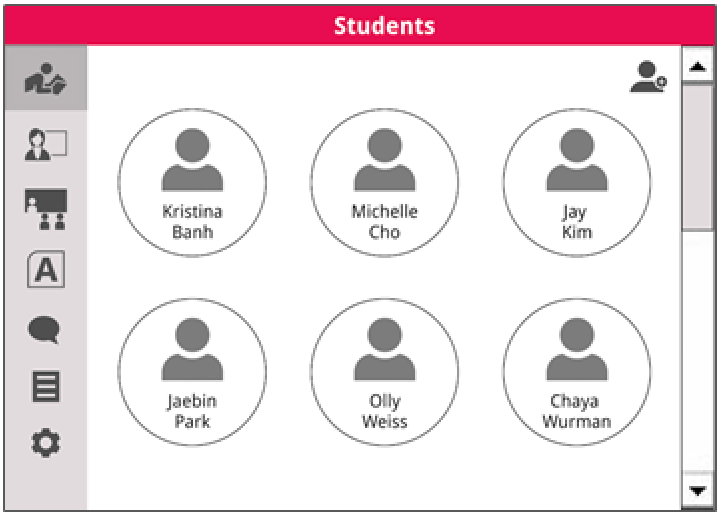
Chava forgot her field trip clothes. Bringing it now.

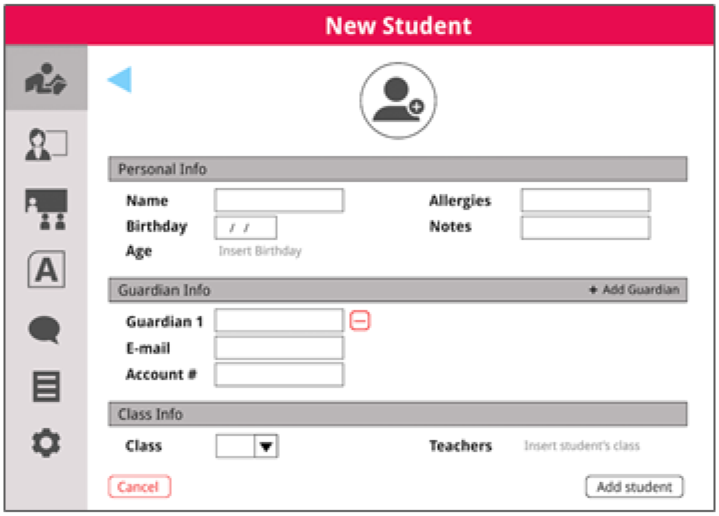
- Rob

Admin Screens

The admin screens give focus on administrative tasks like the ability to edit, add, and remove accounts.

The screens below illustrate how an admin can review a student roster, update student profile information, and remove/add student accounts.





New Student

Personal Info

Name		Allergies	
Birthday	/ /	Notes	
Age	Insert Birthday		

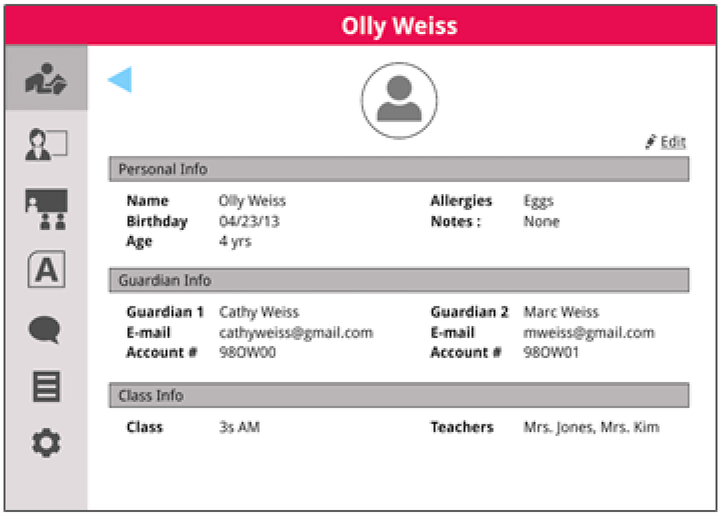
Guardian Info

✚ Add Guardian

Guardian 1	
E-mail	
Account #	

Class Info

Class	▼	Teachers	Insert student's class



Olly Weiss

Personal Info

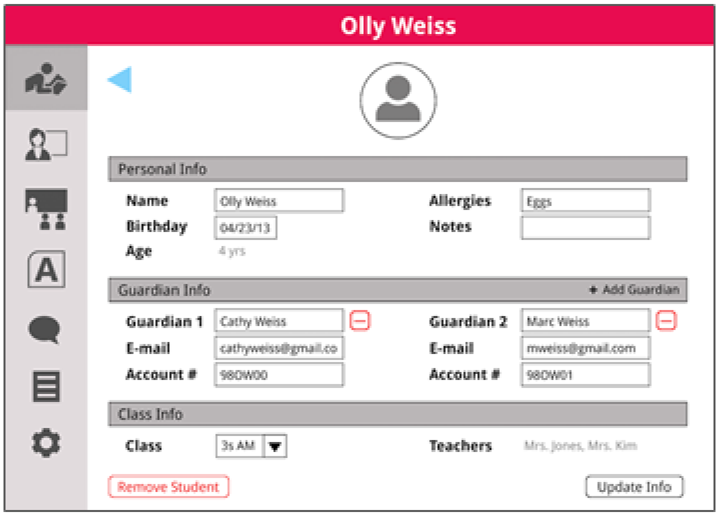
Name	Olly Weiss	Allergies	Eggs
Birthday	04/23/13	Notes :	None
Age	4 yrs		

Guardian Info

Guardian 1	Cathy Weiss	Guardian 2	Marc Weiss
E-mail	cathyweiss@gmail.com	E-mail	mweiss@gmail.com
Account #	98OW00	Account #	98OW01

Class Info

Class	3s AM	Teachers	Mrs. Jones, Mrs. Kim



Olly Weiss

Personal Info

Name	Olly Weiss	Allergies	Eggs
Birthday	04/23/13	Notes	
Age	4 yrs		

Guardian Info

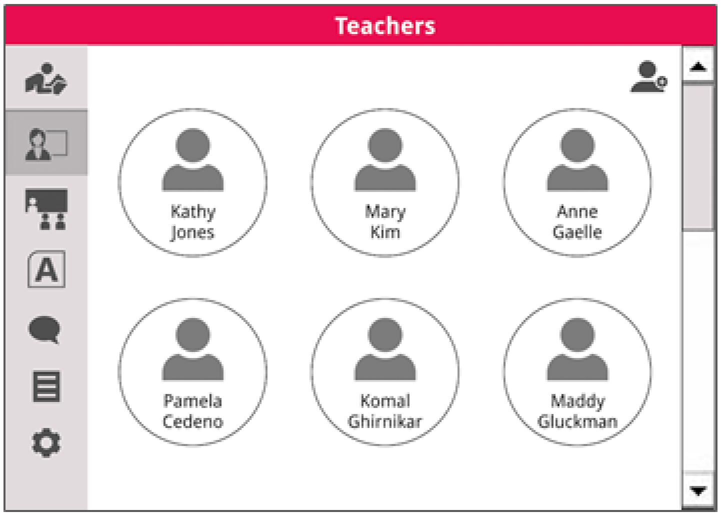
✚ Add Guardian

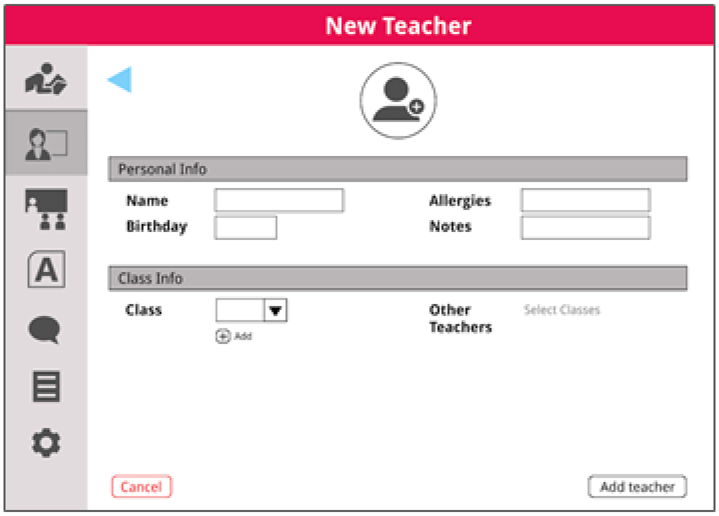
Guardian 1	Cathy Weiss	Guardian 2	Marc Weiss
E-mail	cathyweiss@gmail.co	E-mail	mweiss@gmail.com
Account #	98OW00	Account #	98OW01

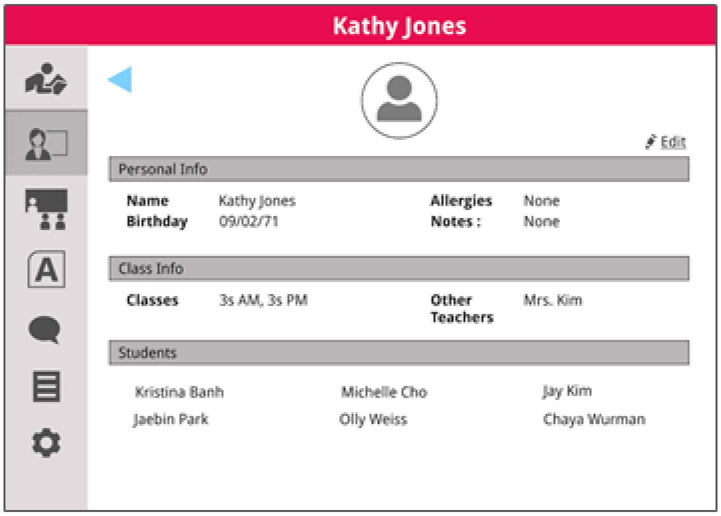
Class Info

Class	3s AM ▼	Teachers	Mrs. Jones, Mrs. Kim

The admin can adjust teacher information in the same way, as seen below.







Kathy Jones

Edit

Personal Info

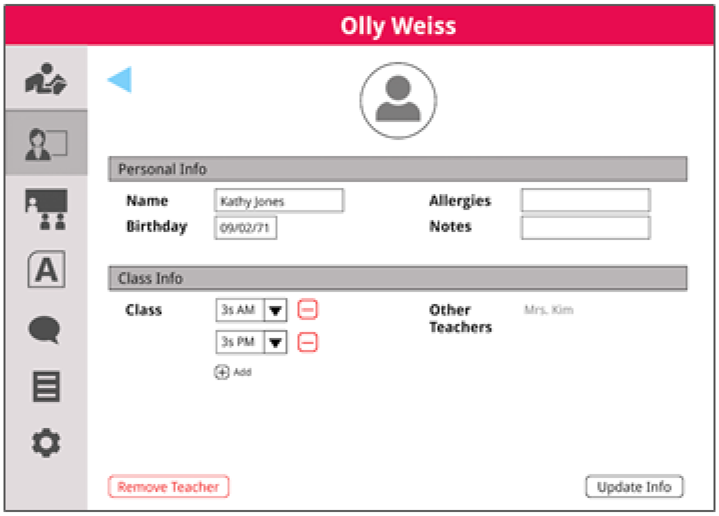
Name	Kathy Jones	Allergies	None
Birthday	09/02/71	Notes :	None

Class Info

Classes	3s AM, 3s PM	Other Teachers	Mrs. Kim

Students

Kristina Banh	Michelle Cho	Jay Kim
Jaebin Park	Olly Weiss	Chaya Wurman



Olly Weiss

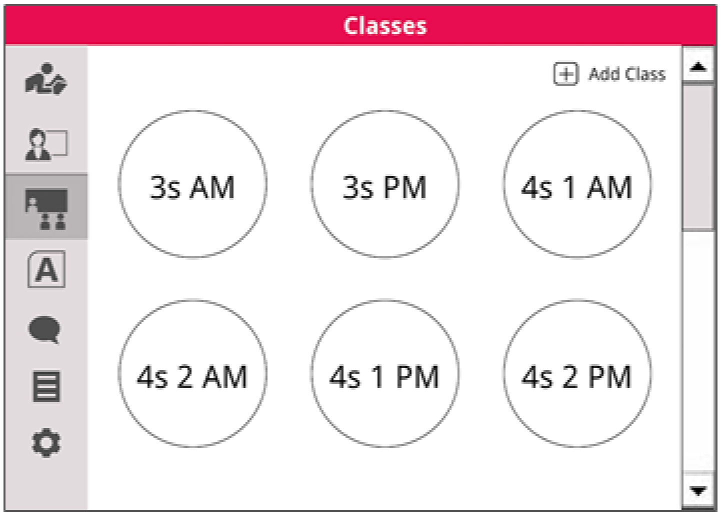
Personal Info

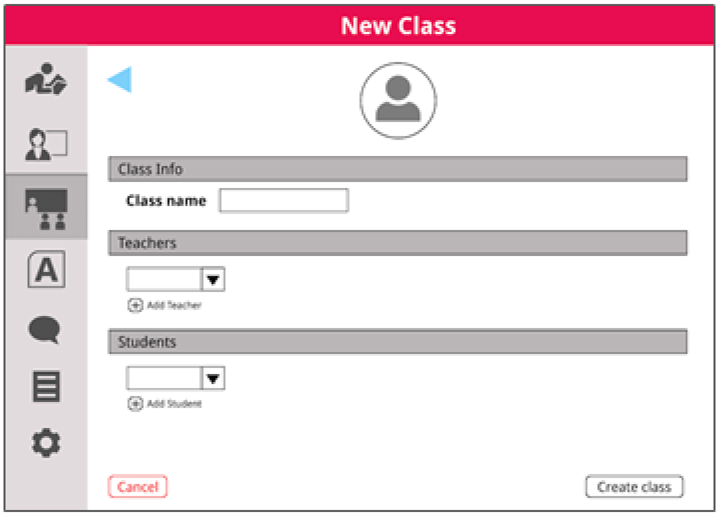
Name	Kathy Jones	Allergies	
Birthday	09/02/71	Notes	

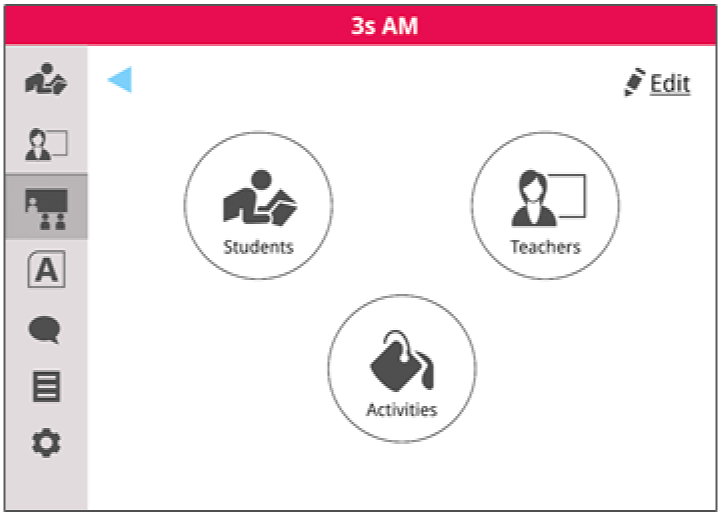
Class Info

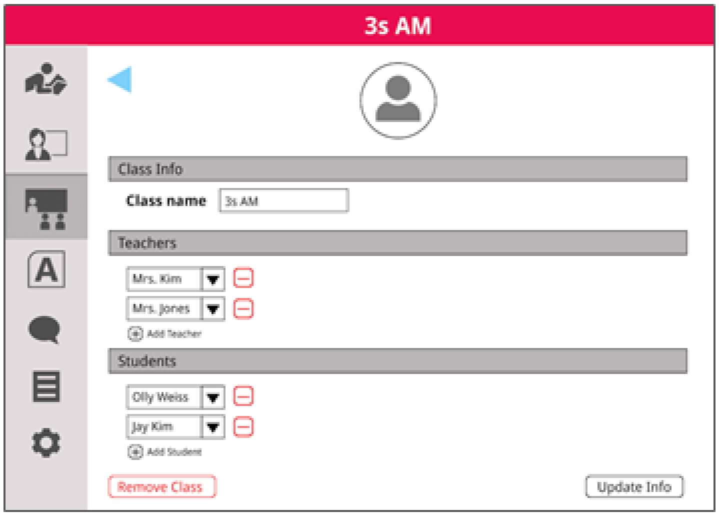
Class	3s AM 3s PM	Other Teachers	Mrs. Kim

In addition, the admin can create and adjust class rosters and actitivities.


Classes
Add Class
3s AM
3s PM
4s 1 AM
4s 2 AM
4s 1 PM
4s 2 PM


New Class
Class Info
Class name
Teachers
Add Teacher
Students
Add Student
Cancel
Create class


3s AM
Edit
Students
Teachers
Activities



3s AM

Class Info

Class name 3s AM

Teachers

Mrs. Kim ▼ ⊖

Mrs. Jones ▼ ⊖

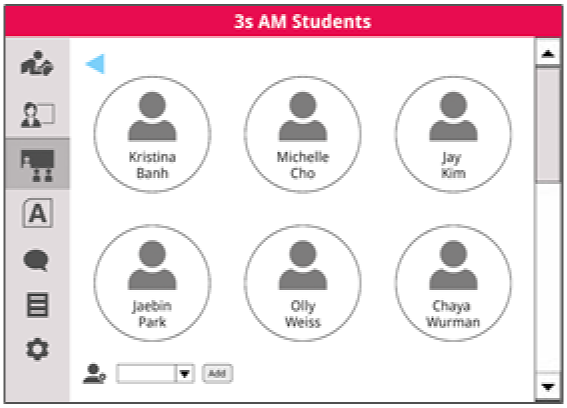
⊕ Add Teacher

Students

Olly Weiss ▼ ⊖

Jay Kim ▼ ⊖

⊕ Add Student



3s AM Students

Kristina Banh

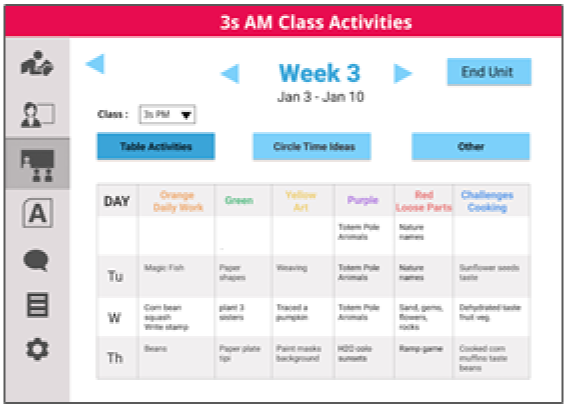
Michelle Cho

Jay Kim

Jaebin Park

Olly Weiss

Chaya Wurman



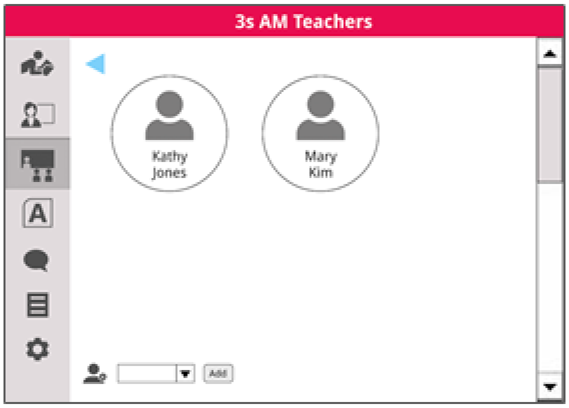
3s AM Class Activities

Week 3

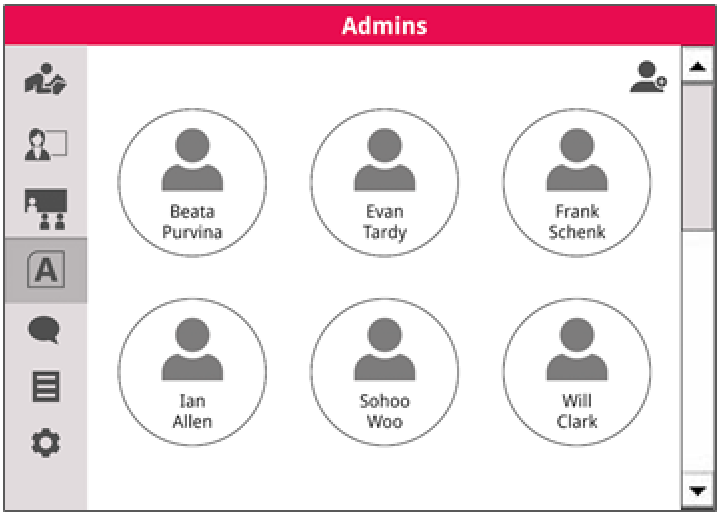
Jan 3 - Jan 10

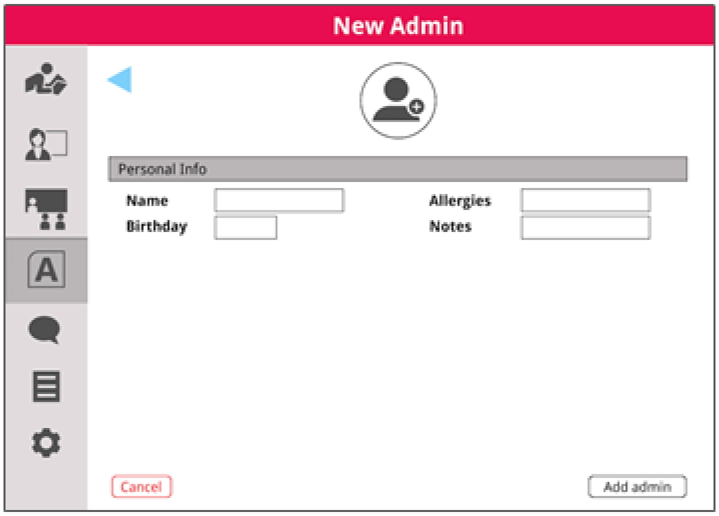
Class : 3s PM ▼

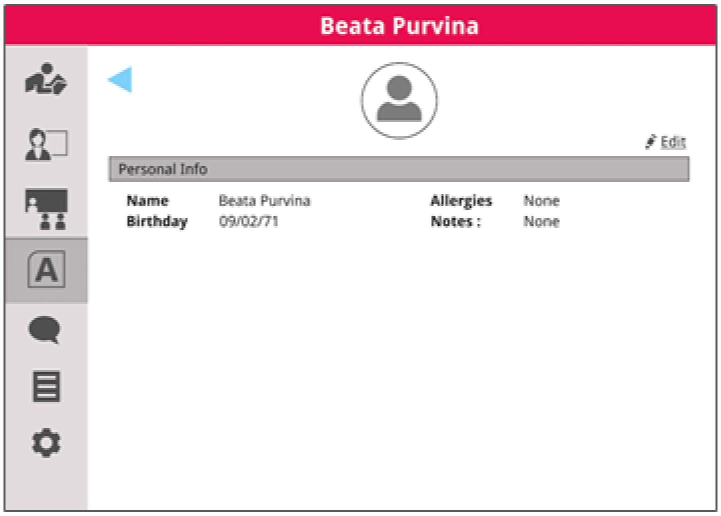
DAY	Orange Daily Work	Green	Yellow Art	Purple	Red Loose Parts	Challenges Cooking
				Totem Pole Animals	Nature names	
Tu	Magic Fish	Paper shapes	Weaving	Totem Pole Animals	Nature names	Sunflower seeds taste
W	Corn bean squash Write stamp	plant 3 sisters	Traced a pumpkin	Totem Pole Animals	Sand, gems, flowers, rocks	Dehydrated taste fruit veg
Th	Beans	Paper plate tipi	Paint masks background	H2O colo sunsets	Ramp game	Cooked corn muffins taste beans



The admin can also add or remove additional admin accounts as needed.



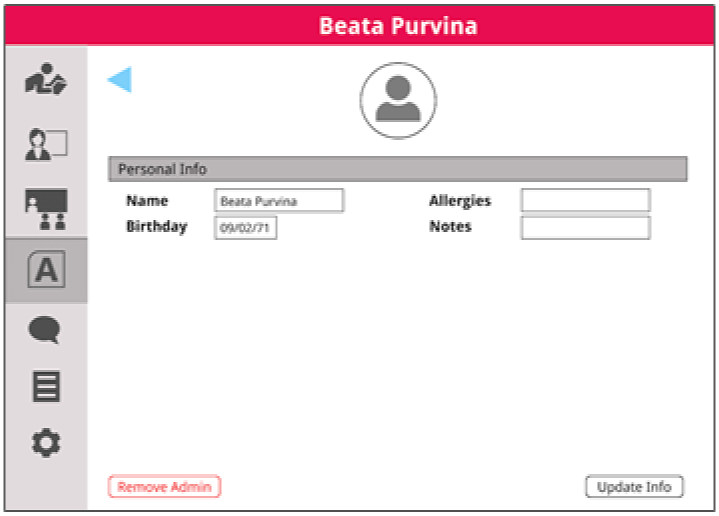




Beata Purvina

Personal Info

Name Beata Purvina
Birthday 09/02/71
Allergies None
Notes : None

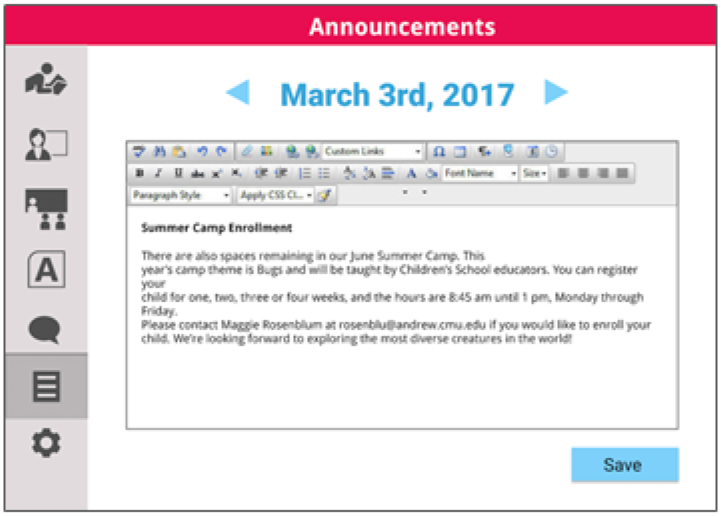


Beata Purvina

Personal Info

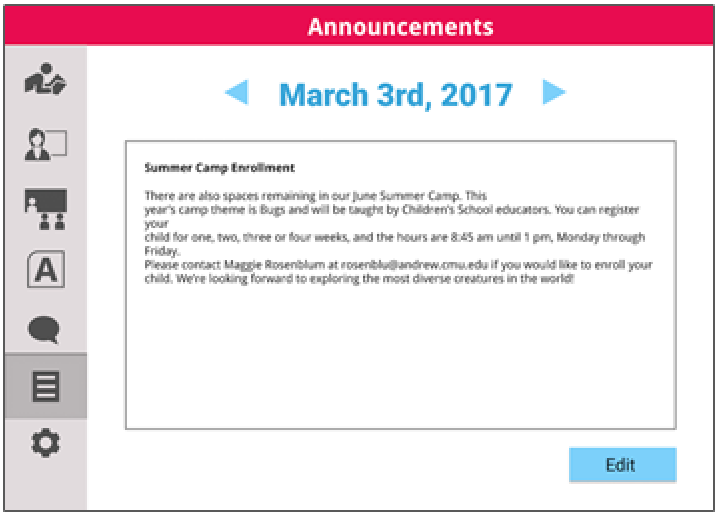
Name Beata Purvina
Birthday 09/02/71
Allergies
Notes

Finally, the admin can send out announcements from these screens.



Summer Camp Enrollment

There are also spaces remaining in our June Summer Camp. This
year's camp theme is Bugs and will be taught by Children's School educators. You can register
your
child for one, two, three or four weeks, and the hours are 8:45 am until 1 pm, Monday through
Friday.
Please contact Maggie Rosenblum at rosenblu@andrew.cmu.edu if you would like to enroll your
child. We're looking forward to exploring the most diverse creatures in the world!



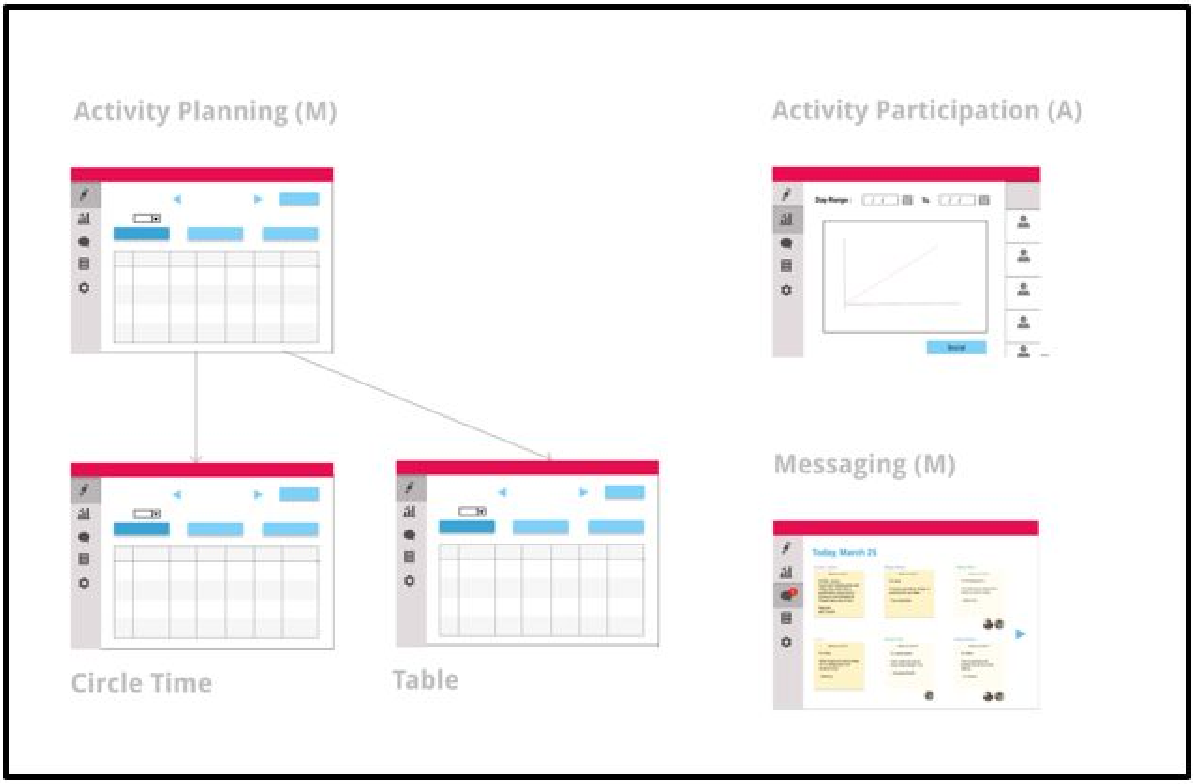
Summer Camp Enrollment

There are also spaces remaining in our June Summer Camp. This
year's camp theme is Bugs and will be taught by Children's School educators. You can register
your
child for one, two, three or four weeks, and the hours are 8:45 am until 1 pm, Monday through
Friday.
Please contact Maggie Rosenblum at rosenblu@andrew.cmu.edu if you would like to enroll your
child. We're looking forward to exploring the most diverse creatures in the world!

Teacher's Screens

Teachers can plan and schedule their class activities through these initial screens. These designs were inspired by the paper templates the Children's School teachers already use.

Website (Desktop)



Activity Planning (M)

Activity Participation (A)

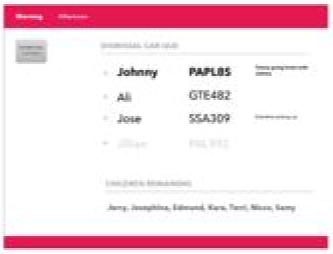
Messaging (M)

Circle Time

Table

Display

Dismissal (A)



Johnny PAPL85

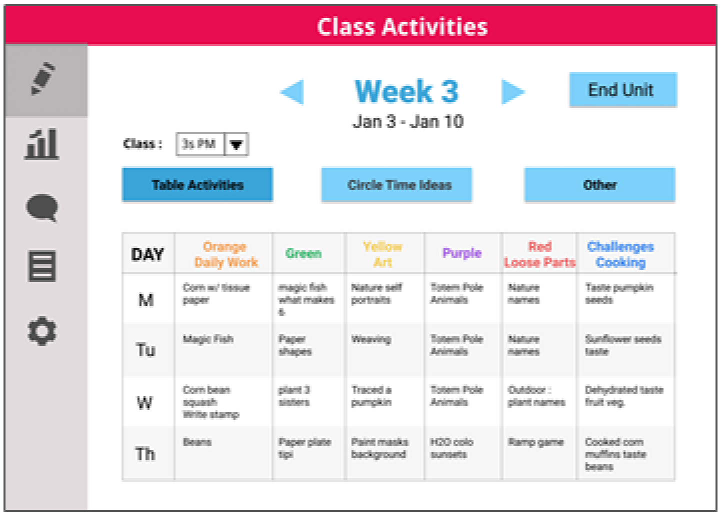
Ali GTE482

Jose SSA309

Jerry, Josephina, Edmund, Kara, Terri, Nissa, Samy

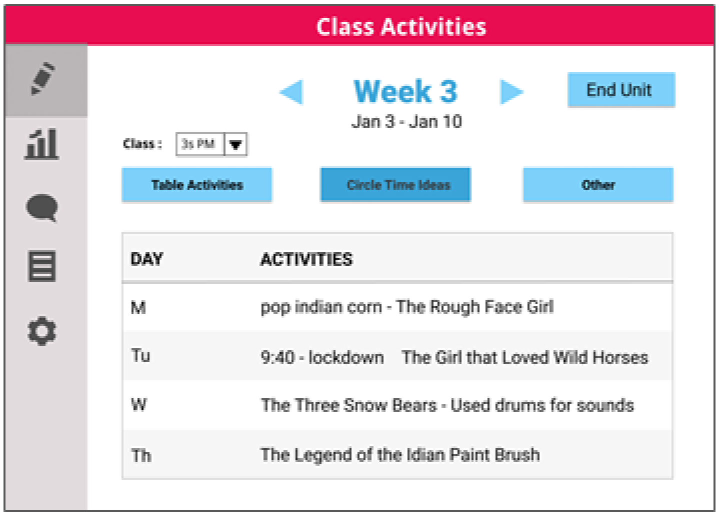
Messaging (A)

Today, March 25



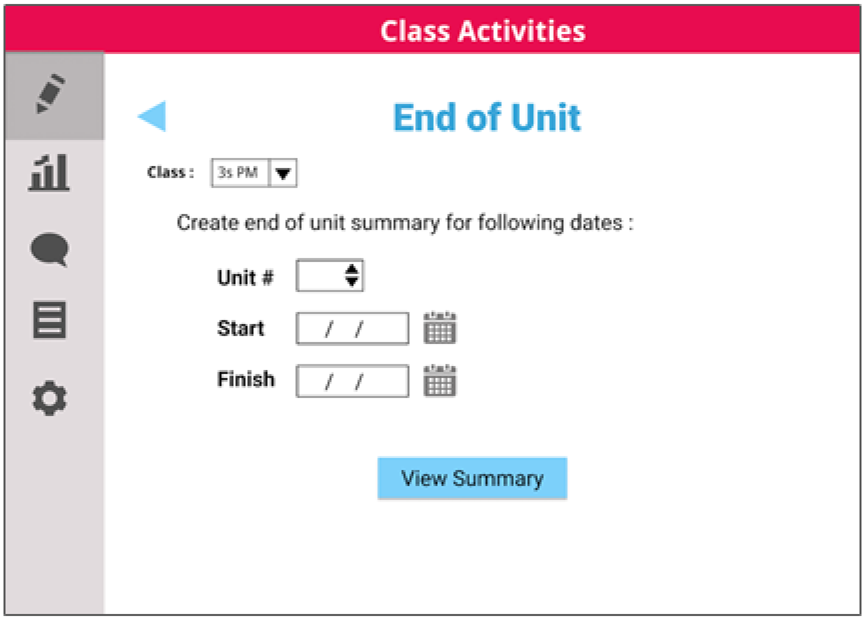
Class Activities

DAY	Orange Daily Work	Green	Yellow Art	Purple	Red Loose Parts	Challenges Cooking
M	Corn w/ tissue paper	magic fish what makes 6	Nature self portraits	Totem Pole Animals	Nature names	Taste pumpkin seeds
Tu	Magic Fish	Paper shapes	Weaving	Totem Pole Animals	Nature names	Sunflower seeds taste
W	Corn bean squash Write stamp	plant 3 sisters	Traced a pumpkin	Totem Pole Animals	Outdoor : plant names	Dehydrated taste fruit veg.
Th	Beans	Paper plate tipi	Paint masks background	H2O colo sunsets	Ramp game	Cooked corn muffins taste beans



Class : 3s PM

DAY	ACTIVITIES
M	pop indian corn - The Rough Face Girl
Tu	9:40 - lockdown The Girl that Loved Wild Horses
W	The Three Snow Bears - Used drums for sounds
Th	The Legend of the Idian Paint Brush



Class : 3s PM

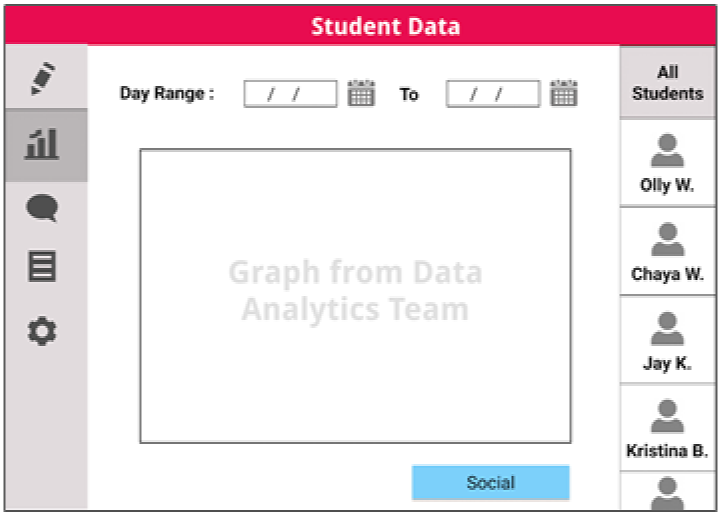
Create end of unit summary for following dates :

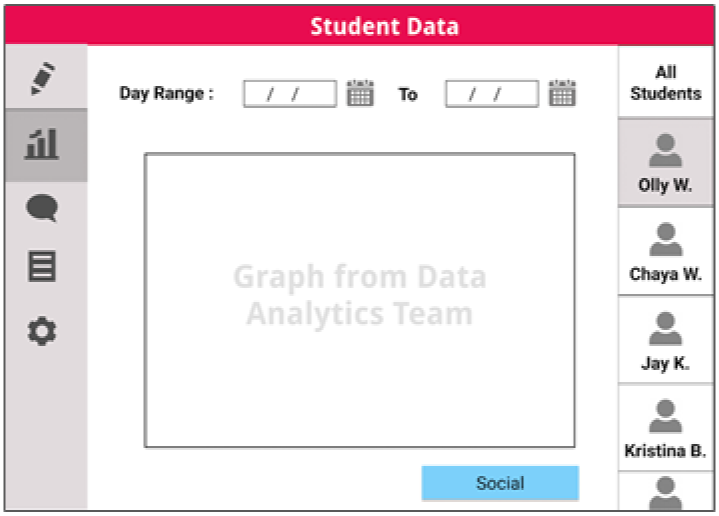
Unit #

Start / /

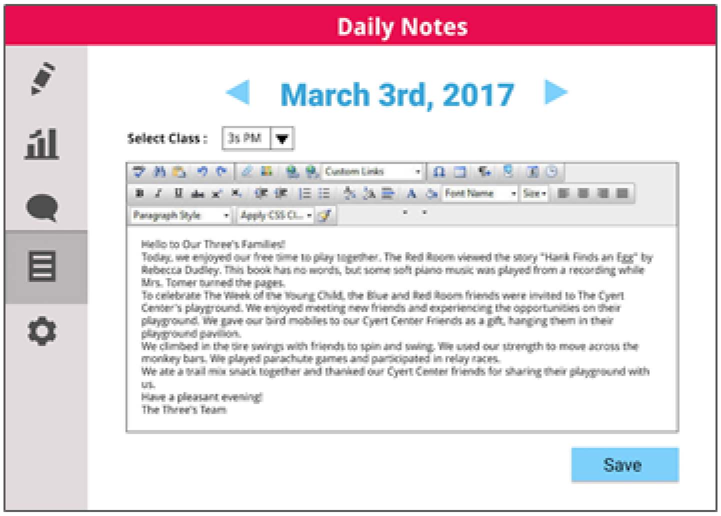
Finish / /

As the Magpie ecosystem collects data, teachers can refer to the following data screens. These pages allow teachers to quickly review student data from different perspectives, helping them make better decisions in the classroom.

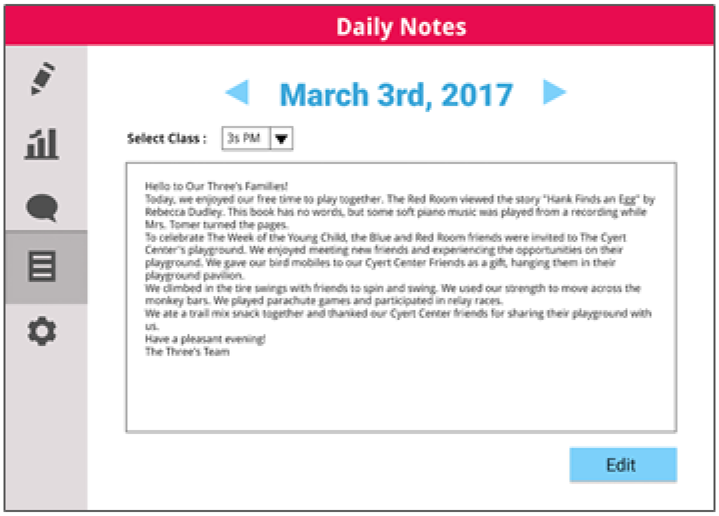




Using this screen, teachers can send parents daily notes describing the day’s activities.



Hello to Our Three's Families!
Today, we enjoyed our free time to play together. The Red Room viewed the story "Hank Finds an Egg" by Rebecca Dudley. This book has no words, but some soft piano music was played from a recording while Mrs. Tomer turned the pages.
To celebrate The Week of the Young Child, the Blue and Red Room friends were invited to The Cyert Center's playground. We enjoyed meeting new friends and experiencing the opportunities on their playground. We gave our bird mobiles to our Cyert Center Friends as a gift, hanging them in their playground pavilion.
We climbed in the tire swings with friends to spin and swing. We used our strength to move across the monkey bars. We played parachute games and participated in relay races.
We ate a trail mix snack together and thanked our Cyert Center friends for sharing their playground with us.
Have a pleasant evening!
The Three's Team

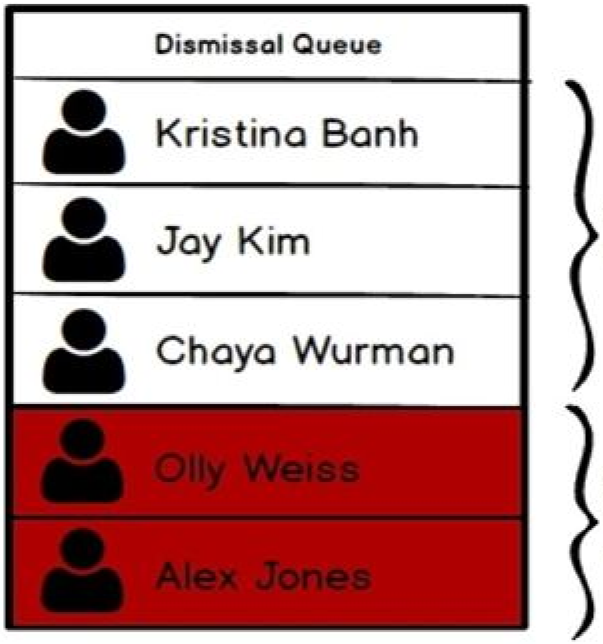


Hello to Our Three's Families!
Today, we enjoyed our free time to play together. The Red Room viewed the story "Hank Finds an Egg" by Rebecca Dudley. This book has no words, but some soft piano music was played from a recording while Mrs. Tomer turned the pages.
To celebrate The Week of the Young Child, the Blue and Red Room friends were invited to The Cyert Center's playground. We enjoyed meeting new friends and experiencing the opportunities on their playground. We gave our bird mobiles to our Cyert Center Friends as a gift, hanging them in their playground pavilion.
We climbed in the tire swings with friends to spin and swing. We used our strength to move across the monkey bars. We played parachute games and participated in relay races.
We ate a trail mix snack together and thanked our Cyert Center friends for sharing their playground with us.
Have a pleasant evening!
The Three's Team

Dismissal Display

This screen allows teachers to easily organize students for dismissal, as parents move through the pickup line. This screen would be accessed through a display screen available in the hallway.

Would be on display in main hallway

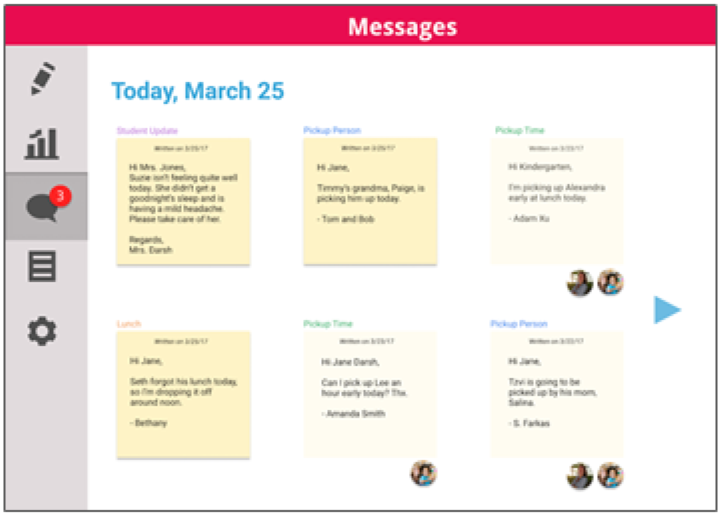


Students already in a waiting area

Students not yet in waiting areas
Once teacher sends student, tap name to make not red

Messaging Screens

This is the high fidelity screen of the teacher view of the parent message board. This display was inspired by the physical messaging board currently used in classrooms at the Children's School. The date is announced at the top of the screen. Post-its frame message text and indicate whether or not a message has been read (pale) or is unread (dark). Colored frames around each post-it indicate message categories. Icons beneath post-its are teachers' profile pictures, and if a profile picture appears beneath a post-it, that means the corresponding teacher has read that message. A scrollbar and blue arrow help teachers view more messages if there are more that are relevant for.



Messages

Today, March 25

Student Update
Written on 3/25/17
Hi Mrs. Jones,
Suzie isn't feeling quite well today. She didn't get a goodnight's sleep and is having a mild headache. Please take care of her.
Regards,
Mrs. Darsh

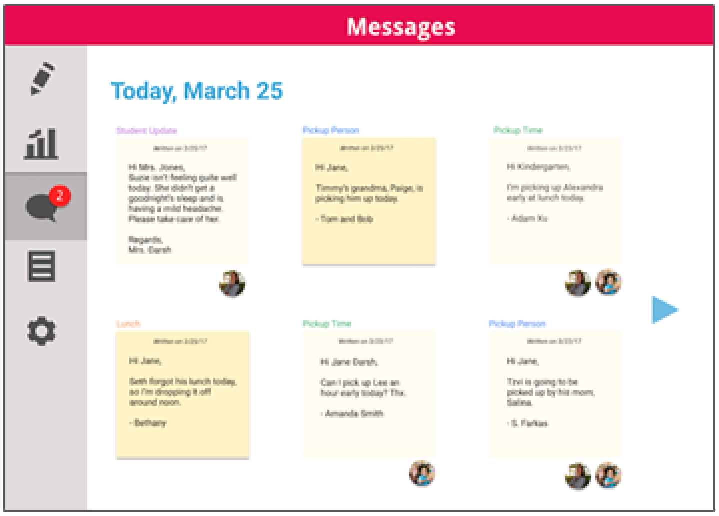
Pickup Person
Written on 3/25/17
Hi Jane,
Timmy's grandma, Paige, is picking him up today.
- Tom and Bob

Pickup Time
Written on 3/25/17
Hi Kindergarten,
I'm picking up Alexandra early at lunch today.
- Adam Xu

Lunch
Written on 3/25/17
Hi Jane,
Seth forgot his lunch today, so I'm dropping it off around noon.
- Bethany

Pickup Time
Written on 3/25/17
Hi Jane Darsh,
Can I pick up Lee an hour early today? Thx.
- Amanda Smith

Pickup Person
Written on 3/25/17
Hi Jane,
Tzvi is going to be picked up by his mom, Salina.
- S. Farkas



Messages

Today, March 25

Student Update
Written on 3/25/17
Hi Mrs. Jones,
Suzie isn't feeling quite well today. She didn't get a goodnight's sleep and is having a mild headache. Please take care of her.
Regards,
Mrs. Darsh

Pickup Person
Written on 3/25/17
Hi Jane,
Timmy's grandma, Paige, is picking him up today.
- Tom and Bob

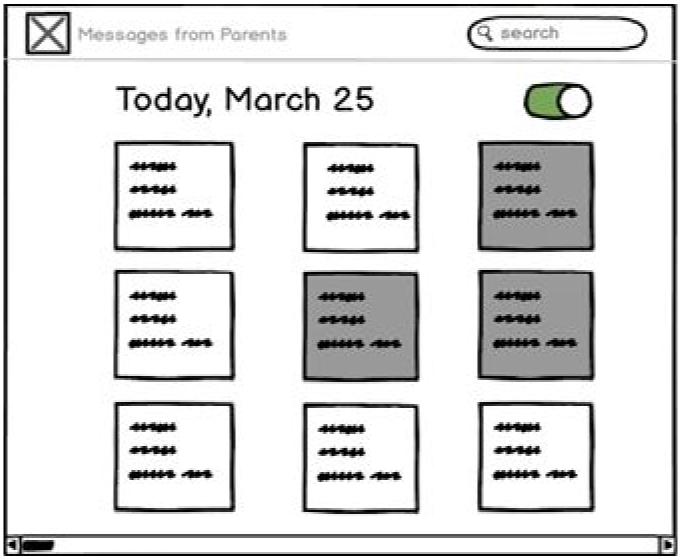
Pickup Time
Written on 3/25/17
Hi Kindergarten,
I'm picking up Alexandra early at lunch today.
- Adam Xu

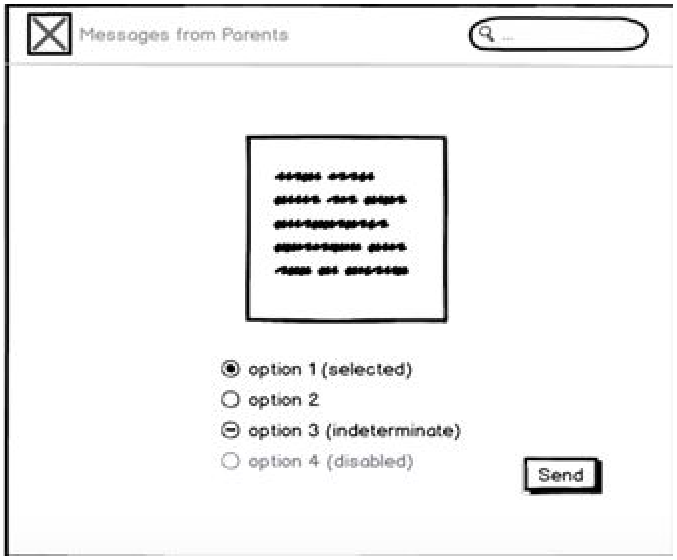
Lunch
Written on 3/25/17
Hi Jane,
Seth forgot his lunch today, so I'm dropping it off around noon.
- Bethany

Pickup Time
Written on 3/25/17
Hi Jane Darsh,
Can I pick up Lee an hour early today? Thx.
- Amanda Smith

Pickup Person
Written on 3/25/17
Hi Jane,
Tzvi is going to be picked up by his mom, Salina.
- S. Farkas

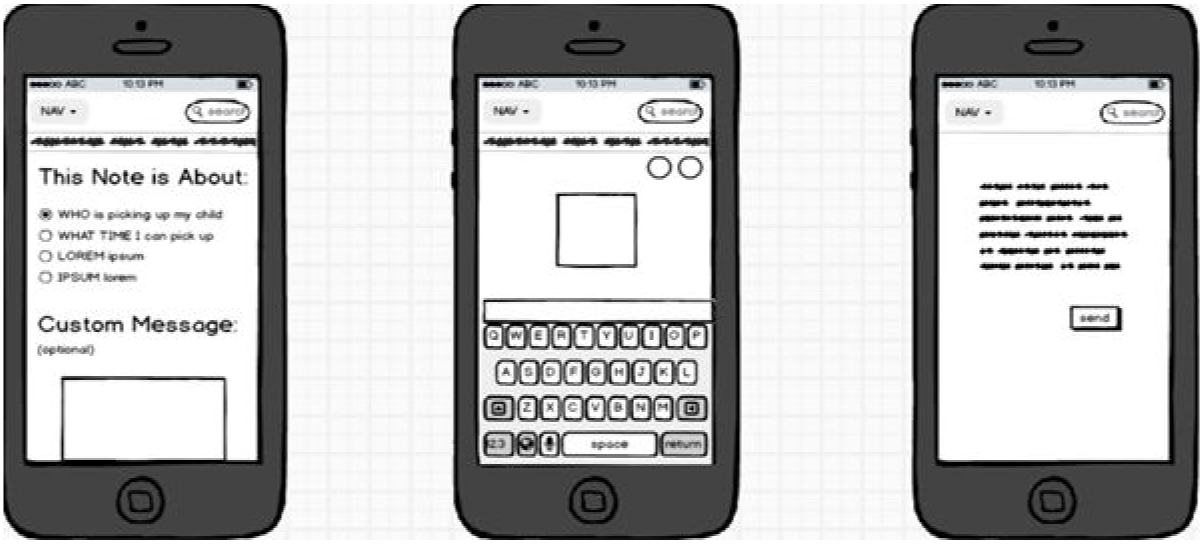
These are medium fidelity wireframes. The top image is the wireframe that corresponds to the previously shown teacher view of the parent messaging board. The bottom image is a response screen option for teachers so that they can quickly select from default response options to send to parents. The two default messages will be: “Got it, thanks!” and “Please contact me via email or see me in person about this note.”





● option 1 (selected)
○ option 2
⊖ option 3 (indeterminate)
○ option 4 (disabled)

These are medium fidelity wireframes of the parent mobile app messaging capabilities. Here, we see a screen showing that parents can select the category into which their message falls from a multiple choice menu. We also see screens showing how a parent would input their message and then preview their message before sending. The bottom left screen shows what a parent would see upon viewing their previously sent messages. There will also be a screen showing how a parent would see a teacher's quick response to their message.



4.2.4 Experimental Evaluation

Teachers

Our research uncovered the activity planning processes that are already in place at the Children's School. This informed our design, as we modeled our interfaces off of their current offline processes. For example, our forms and color-coding are very similar to the paper documents teachers already fill out. The teachers we interviewed expressed a great deal of excitement in having these processes taken online.

Our initial designs focused on messaging within a desktop experience, but our prototyping sessions indicated that this would not be the best solution. Many teachers handle messages, so they desired a solution that would allow them to reviews notes on a more ad hoc basis. They also mentioned that they do not use desktops often in the classroom, which ultimately led us to focus our efforts on a display.

Finally the teachers described a desire for categorized messages from parents, which would make it easier sort and capture information.

Parents

Our initial designs focused on simple messaging and activity tracking screens. Our prototyping session showed us that there would be value in providing additional information. The parents described their challenges tracking down the monthly newsletter and Children's School website.

The parents were particularly excited by having the information they need consolidated in a single place. This spurred us to add the “resources” page in the final designs.

The parents also indicated a strong desire for a mobile experience, as they almost always engage with Children’s School content on their phones.

Admins

Our research indicated the need for flexibility as things evolve at the Children’s School. The admin screens were designed to have simple functionality that can quickly add or remove groups, add or remove users, and send messages to the entire school.

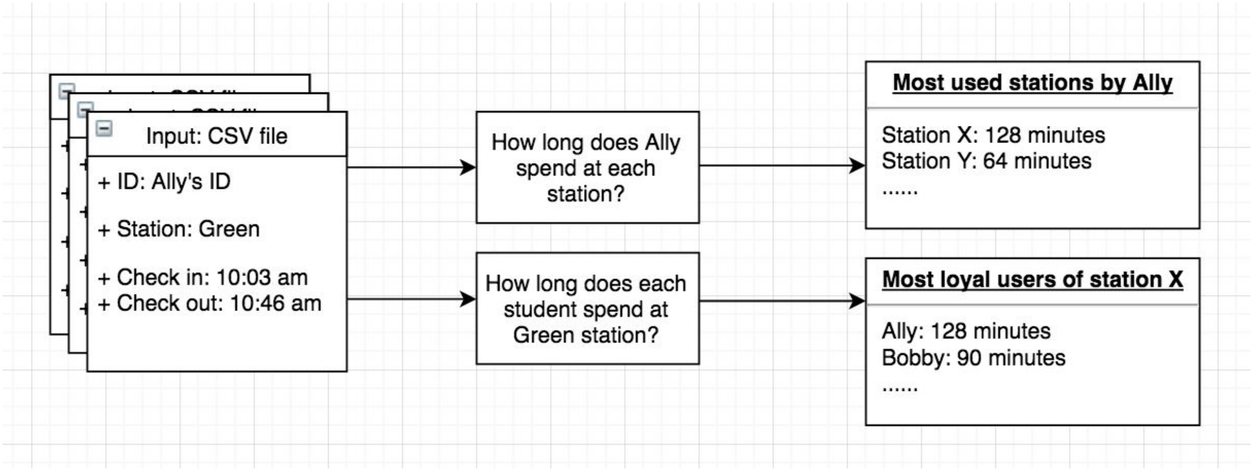
4.3 Data Analytics

4.3.1 Functionality

Functionality was set so that other groups could be able to request and retrieve the relevant data analytics that would be important for their group to display. The script was designed so that it would be able to output all analytics and plots for a previous day or specified timeframe. The front end group’s application would also be able to access our data processing script and also call individual functions within our script.

4.3.2 Interactions

The csv data file named in a specific datetime format would be taken in as an argument by our script. Then, the main function of the data file would be called where the data file would be parsed into appropriate data structures so that all the functions for generating plots would be called. These plots would be generated and saved in corresponding folders (on the local filesystem) with the appropriate datetime_analytic format (.html), along with some of the data structures that were generated. For analytics involving longer ranges of time in the past, there would be calls for either that past data and/or data structures containing it. This data would be combined with the other data structures previously created.



4.3.3 Algorithms

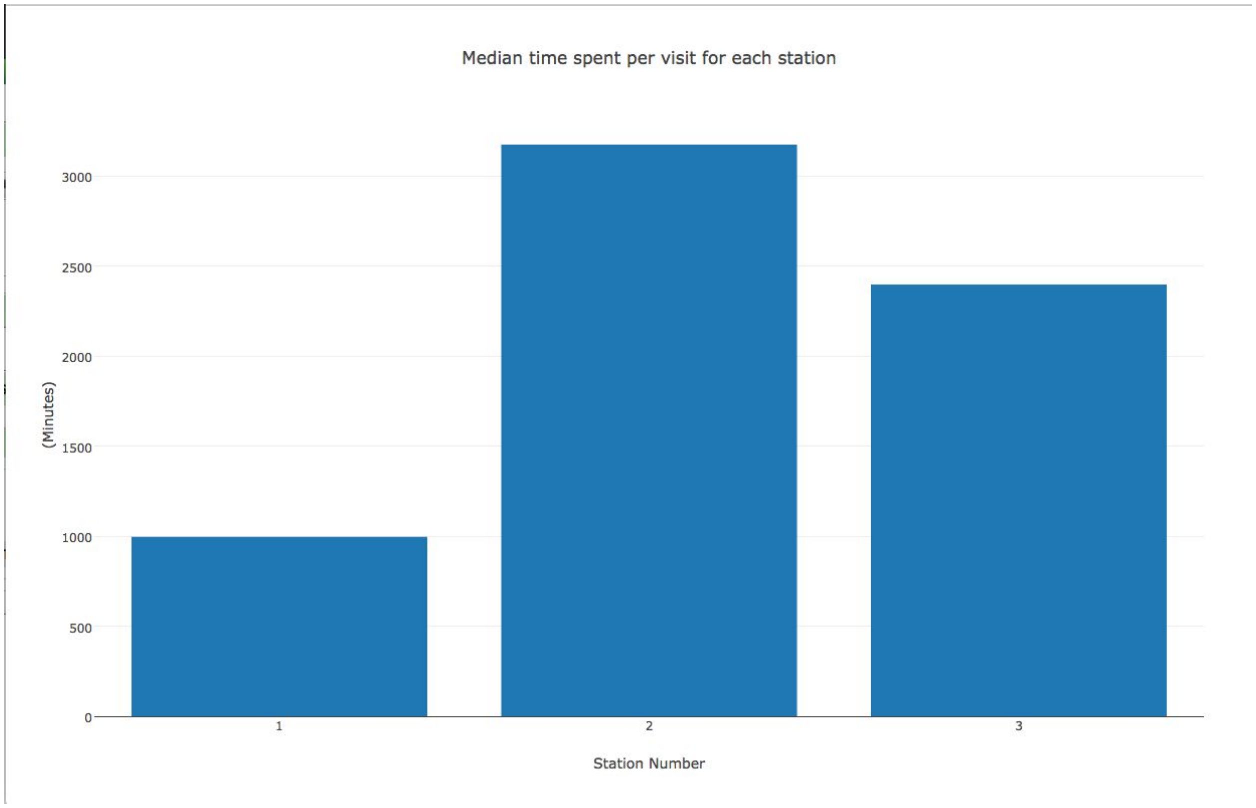
Algorithms for analytics were broken down into two sections - the student and the station. For the student, analytics included average time per visit per station, the total amount of time spent at a station, and the top visited station given a inputted time period. For the station, analytics included (from all students) average time per visit per station, total amount of time spent at a station, median amount of time spent including all visits per station, and plots of the time of each visit by each student.

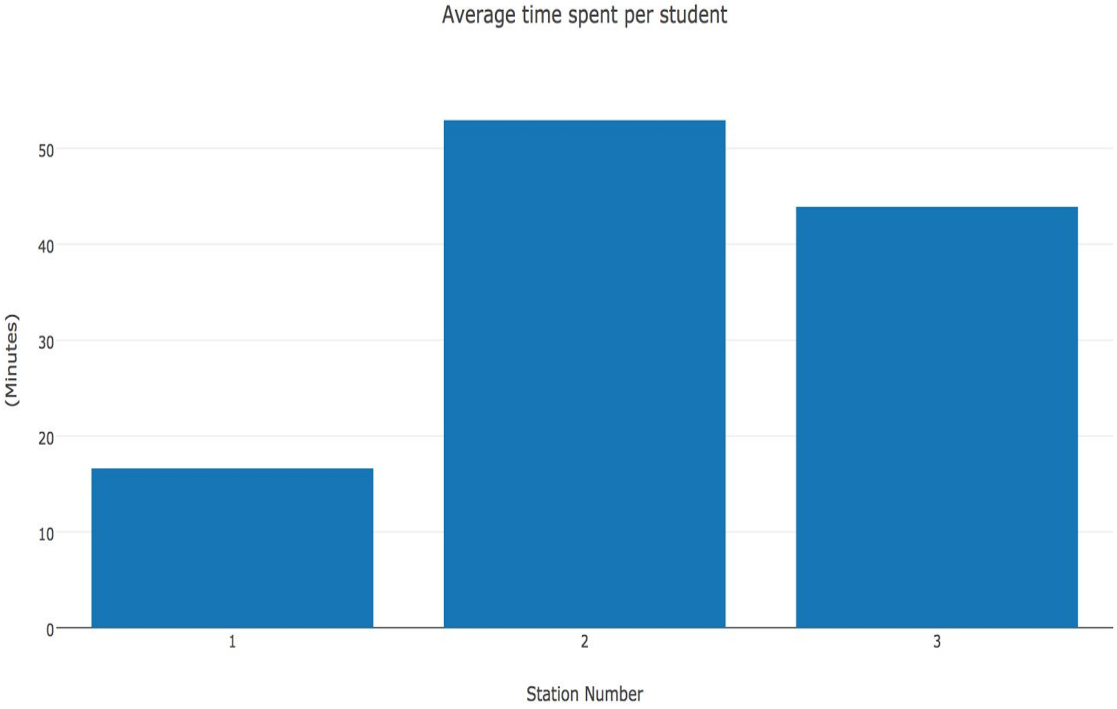
These analytics were generated through parsing the data into specific classes and intermediate data structures defined by our script; the code was organized in an object-orientated fashion. Structuring the data in an object-orientated fashion helped break down/modularize our code and thus make the analysis more intuitive and extensible. The intermediate data structures (baseStationOverview and studentOverview) were classes holding all the relevant data, which would then be passed to the functions actually performing the analysis as arguments. We had several functions to help transform the data into the intermediate data structures that we needed. These intermediate data structures aggregated the data (that was processed into multiple Tables class objects), and made the further analysis easier to perform. After performing the analysis, further pickling (serializing the data objects into an independent data stream) helped save/reuse the data for future analyses. The plots ultimately generated were utilizing Plotly offline and saved in .html format.

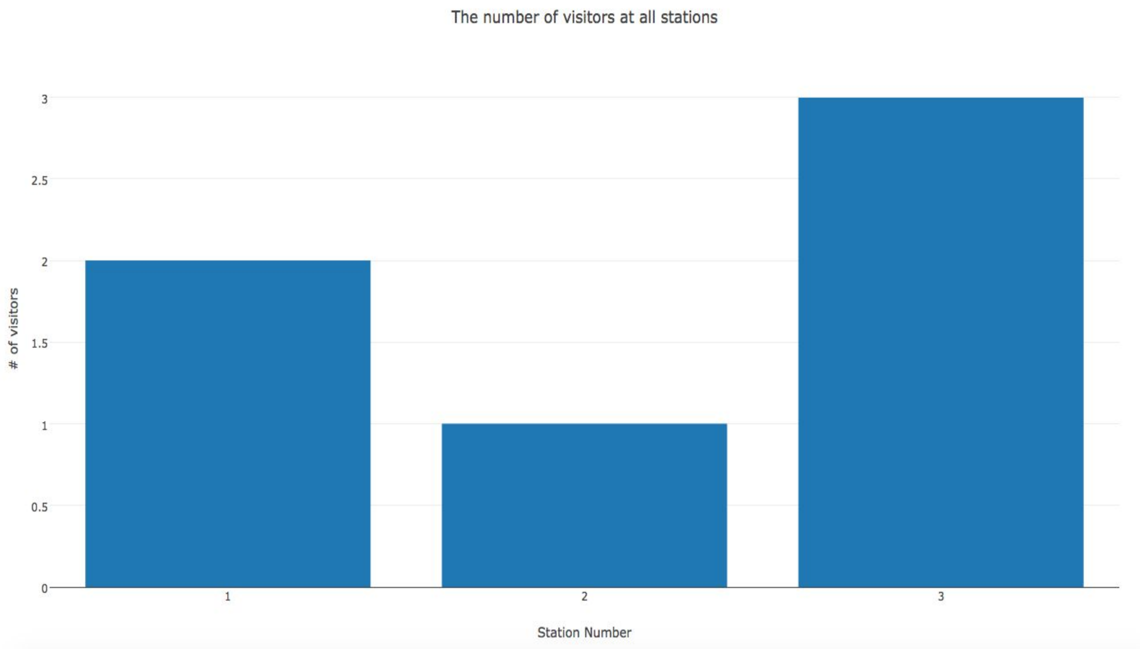
Classes:
Data: tag_id, base_station_id, timestamp, enter_flag
TimeBlock: tag_id, base_station_id, enter_time, leave_time
Table: timeblocks, tagBlocks, tagStations, baseBlocks, baseStudents, baseTotalTime

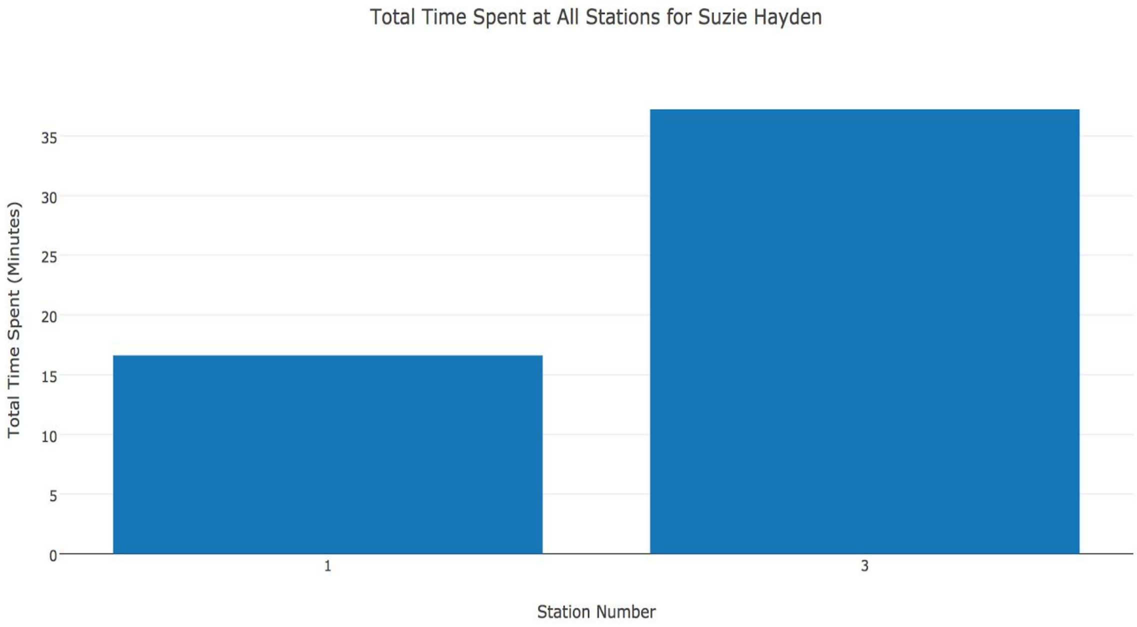
Intermediate Data Structures:
baseStationOverview: start, end, days, daysMap, baseStationIds
studentOverview: start, end, days, daysMap, tagIds

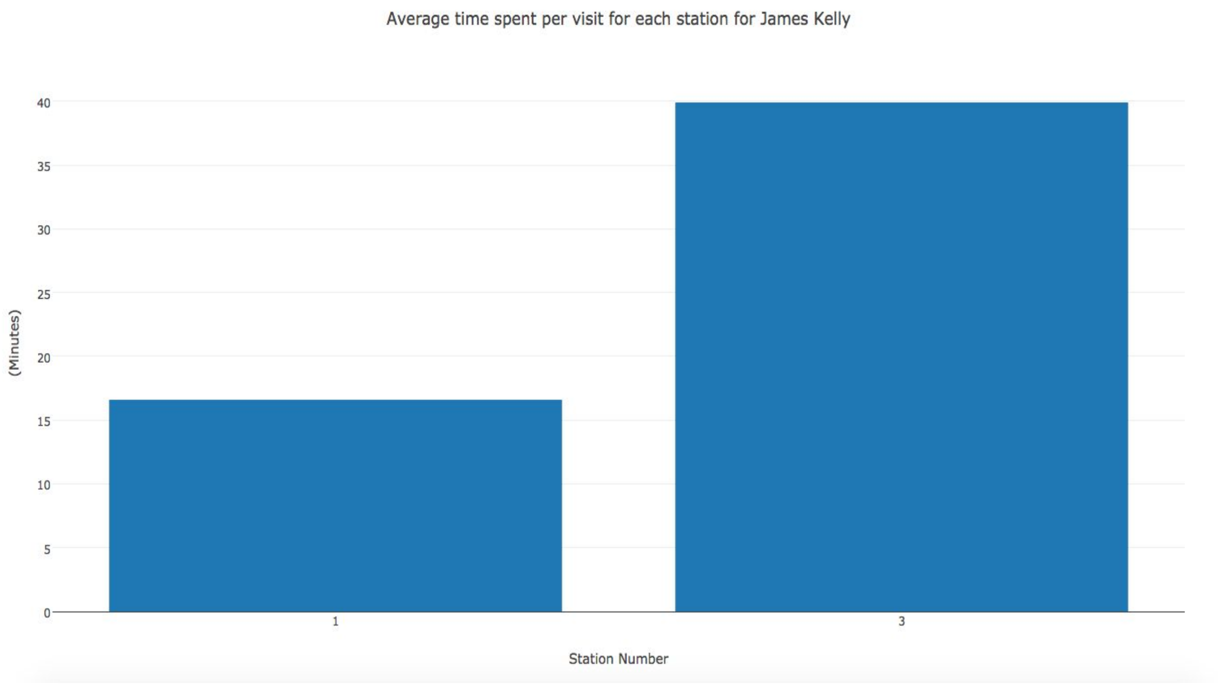
Examples of Charts generated:


Median time spent per visit for each station
(Minutes)
3000
2500
2000
1500
1000
500
0
1
2
3
Station Number


Average time spent per student
(Minutes)
50
40
30
20
10
0
1
2
3
Station Number


The number of visitors at all stations
# of visitors
0
0.5
1
1.5
2
2.5
3
1
2
3
Station Number

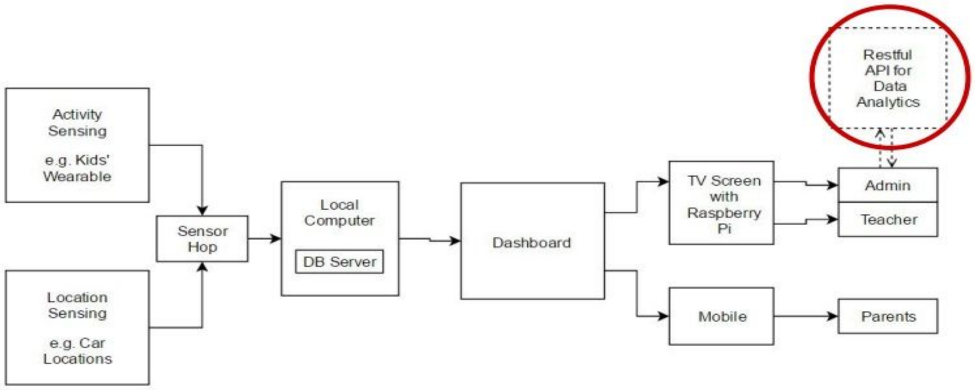

Total Time Spent at All Stations for Suzie Hayden
Total Time Spent (Minutes)
0
5
10
15
20
25
30
35
1
3
Station Number



4.3.4 Experimental Data Set

Activity and sensing team provided us with .csv files as mock data to run and test our code. Our code followed the format that was previously agreed upon between the teams. Unfortunately the format was changed multiple times and it was difficult to accommodate the change in format due to the time constraint. We wrote our own mock data and exhaustively tested our code with the help of the new mock data set. We successfully passed all the tests and created bar graphs for the same.

4.3.5 Software Architecture



In the system architecture diagram, we interact with the frontend team. We receive CSV files from them with raw data, analyze and process that data, generate graphs for the processed data and then send the processed data with the updated statistics and the png files of the graphs generated to the frontend team.

4.3.6 Software Modules and Status

At the end of phase 3, we were able to code and test all of our P0 and P1 priority features. We were able to integrate with the frontend team to develop graphs for the features previously mentioned.

Our code is completely written in Python. We used matplotlib as well as as Plotly to generate our graphs for data visualization for the frontend team.

4.4 Activity Sensing

4.4.1 Functionality
4.4.2 Interactions
4.4.3 Screens
4.4.4 Experimental Measurements
4.4.5 Software Architecture
4.4.6 Software Modules and Status
4.4.7 List of components, cost, power

4.4 Activity Sensing

4.4.1 Functionality

The Activity Sensing group used RFID technology to develop a system that would track students' day-to-day engagement with various activity stations in school. Each activity station will house a passive RFID reader that will detect the passive tags that the children will be wearing in wristbands. The data collected will be processed to discard possible conflicting and nosy pings. It is then sent to the local database for further analysis and visualization.

The choice of passive RFID allows for a battery-free wearable, a quality that is highly desirable since it removes the need for charging and increases its durability. Passive RFID readers, moreover, have a limited range, and this is well-suited for our needs.

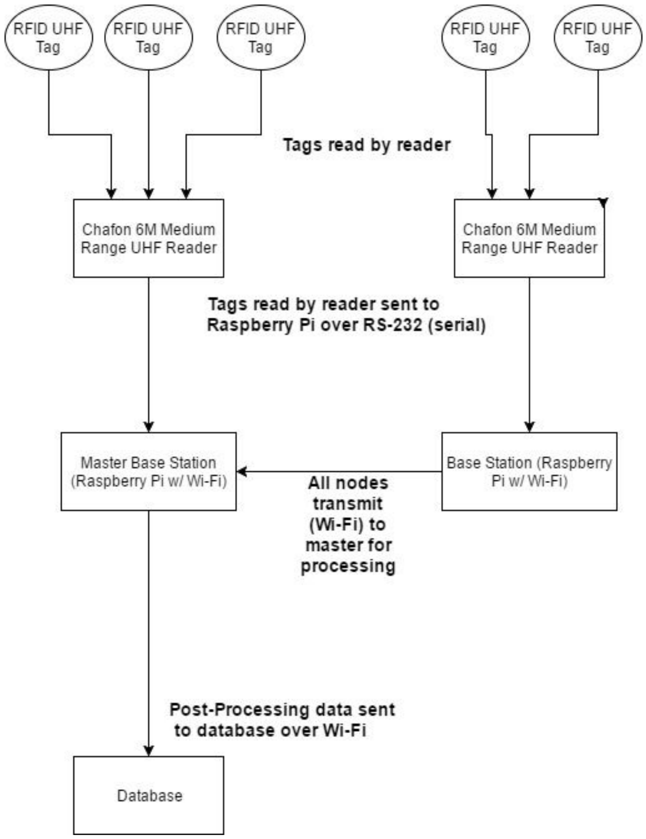
4.4.2 Interactions

The Activity Sensing group interacts with the back end group, which stores the data collected by the RFID readers. A CSV file with the below format will be sent to the database server using an HTTP POST request.

"<time>,<child_tag_number>,<station_id_number>,<entering/leaving>\r\n"
Time format: MM/DD/YYYY HH/MM/SS
Entering = 1, leaving = 0

4.4.3 Hardware Architecture

The passive RFID subsystem requires ultra high frequency (UHF) RFID tags, a 6-meter medium range RFID reader, and Raspberry Pi computers. The hardware components and their interactions are shown in the diagram below:


RFID UHF Tag
RFID UHF Tag
RFID UHF Tag
RFID UHF Tag
RFID UHF Tag
Tags read by reader
Chafon 6M Medium Range UHF Reader
Chafon 6M Medium Range UHF Reader
Tags read by reader sent to Raspberry Pi over RS-232 (serial)
Master Base Station (Raspberry Pi w/ Wi-Fi)
Base Station (Raspberry Pi w/ Wi-Fi)
All nodes transmit (Wi-Fi) to master for processing
Post-Processing data sent to database over Wi-Fi
Database

4.4.4 Software Architecture

Our code primarily was strictly scripted in Python, version 2.7.6.
All data from the different reader nodes will be sent to a single, master node. Each individual reading will contain the following information:

1. Reader Node ID
2. Time stamp (“YYYY-MM-DD hh:mm:ss” format)
3. Tag ID
4. “true” or “false”, Python boolean determining whether Tag “entered” Reader Node’s area

This data will be sent in a serial format, and will be handled by the following classes that we have created.

Class OneTag: This class contains all the individual readings at the current time for a particular tag. This means that it will contain all the readings of this tag pinging of each of the relevant nodes. The nodes will be sorted according to signal strength, enabling us to determine which activity station the child is near or if the child is at none of the stations.

Class AllTags: This class contains the information from all tags. Hence, it is made up of a list or a collection of instances of Class OneTag. This enables us to know where all the children are at a given point of time.

Information is taken from the Class AllTags and is compiled into a comma-separated values (CSV) file once every minute, the desired sampling frequency. The CSV file format per data entry is as follows:

"<time>,<child_tag_number>,<station_id_number>,<entering/leaving>\r\n"
Time format: MM/DD/YYYY HH/MM/SS
Entering = 1, leaving = 0

Once compiled, the CSV file will be sent to the database server using an HTTP POST request, unless the backend team has a different method that it prefers.

4.4.5 Software Modules and Status

An algorithm has been developed in order to pre-process the data and send it to the Database team via a CSV file. The classes described in the section above (4.4.4 Software Architecture) have been created in Python, and have been successfully tested. We have tested the sensitivity of the passive tags and readers to set a proximity threshold to establish whether a tag can be reasonably guaranteed to be near an activity station. We have also figured out how to output data to a CSV file.

4.4.5 List of components, cost, power

Table 3.3.2 Cost and Power of Activity Sensing Hardware

Component	Cost	Power Requirements
Chafon 6M Medium Range UHF Reader	$209.90	+9V DC power supply
RFID Tag	$0.50 - $0.80	N/A
Raspberry Pi 3	$38.27	5V DC power supply, 2.5 amps

Table 3.3.2 Power requirements and cost of hardware components

4.5 Dismissal

4.5.1 Functionality

Synapse Wireless SN132HO-NR

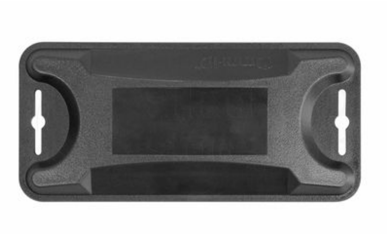
Synapse Wireless SN132HO-NR is an engine that can program a RF chip to be either a reader or a tag. One Synapse Wireless RF200P81 will be programmed using this engine to be an active tag and the other will be an active tag reader. his can be done through Synapse Portal IDE which is a software resource provided by Synapse to configure the module.

Synapse Wireless RF300PD1　　　　*Synapse Wireless RF200P81*

These two Synapse chips are programmed by the Synapse module above to be the reader or the tag. Both require battery or some kind of power source to work. The chips were programmed by Portal IDE, a software resource provided by Synapse.

Synapse RF200 and RF300 active RFID tags were chosen as they can be programmed for use as both tag and reader. The Portal IDE allows to easily configure the RFID tags using simple

Python scripts. The particular models were chosen based on their frequency range in which they operate so that they can be compatible with the passive tag system.

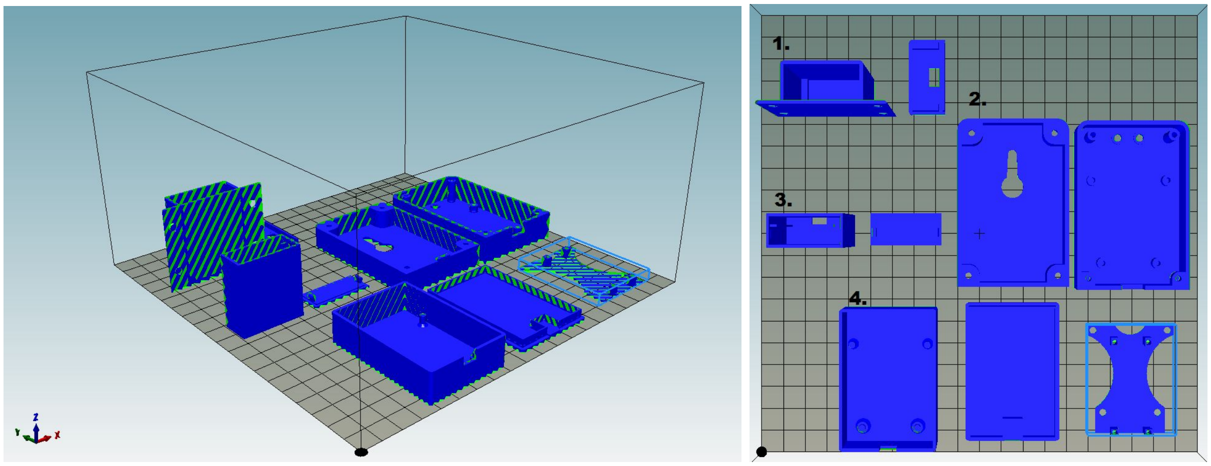


Omni-ID Dura 1500 RFID Tag *Chafon 6M Medium Range UHF Reader Writer*

Omni-ID Dura 1500 RFID Tag is a passive tag that does not require a power source. This is secondary design to be compared with active tag system. The tags do not need to be programmed unlike the active tags above.
The passive RFID tag based system is developed for the final solution. Passive tags are affordable compared to active tags. Therefore, despite the large cost of a single passive tag reader, the complete solution can be more cost effective compared to an active tag based solution.

CanaKit Raspberry Pi 3 Kit with Clear Case and 2.5A Power Supply

Raspberry Pis are easily programmable single board computers and this particular kit supports WIFI and Bluetooth connectivity. It can easily interface with the Synapse RFID reader system over serial communication port to receive data from the tags. The received data will be processed on the Raspberry Pi and converted into HTTP POST request that will be sent to the backend system.

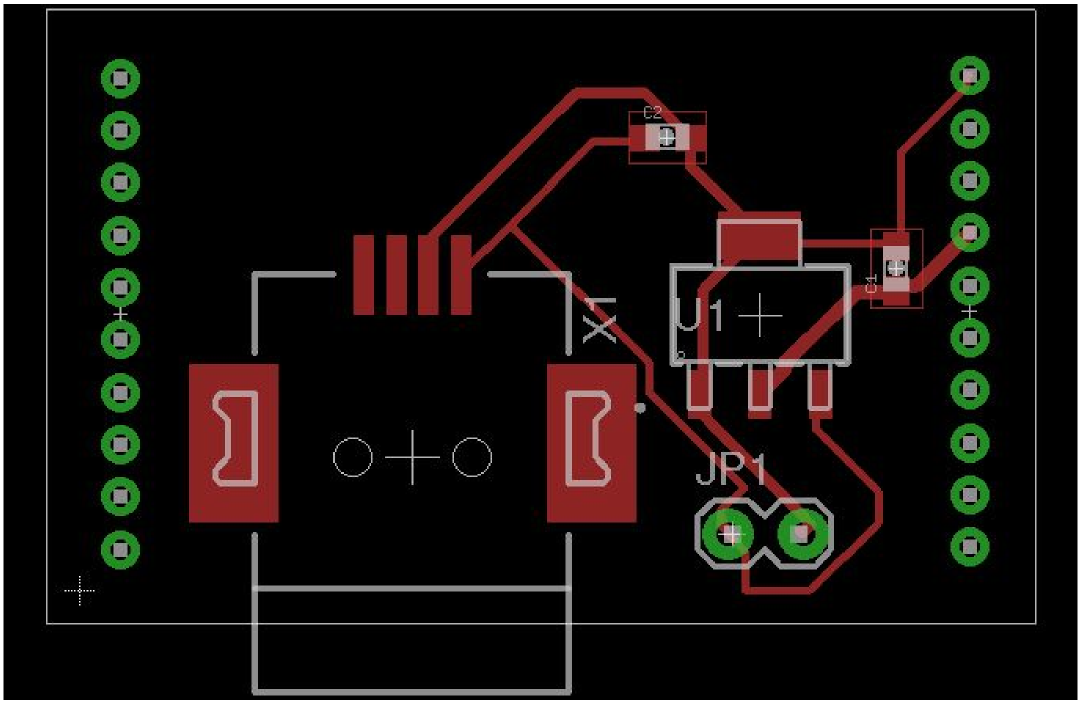


3D Printed Housing Design Options

A customized 3D printed shell will be used to house the reader setup, to increase its lifecycle and protect it from the elements. The design of the reader has following features:

- Internal mounts for reader and raspberry pi
- Option for wall mounting
- Micro USB outlet
- Moisture and dust proof
- Robust

PCB for Car Cigarette Lighter Power Adapter and USB



This contains voltage regulator that linearly steps down the voltage level from cigarette jack or a USB cable down to 3.3 volts. This simple PCB contains headers, 2 capacitors for smooth flow of power.

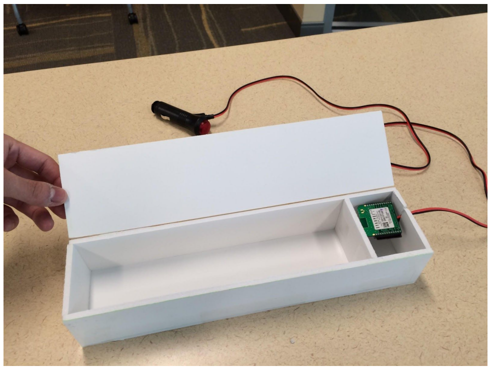
4.5.2 Interactions

For both active tags and the passive tags, the interaction between the tag and the reader will occur once the tag is in range of the reader. The parent will arrive at the dismissal location with preprogrammed tag inside their vehicle. When the tag is inside the range of the reader, the reader will read the tag ID from the tag and that data will be processed on Raspberry Pi that will have the scripts to create the HTTP POST request with the timestamp and tag ID data. Since the Raspberry Pi is connected to WIFI, we can easily transmit the HTTP request to the backend system.

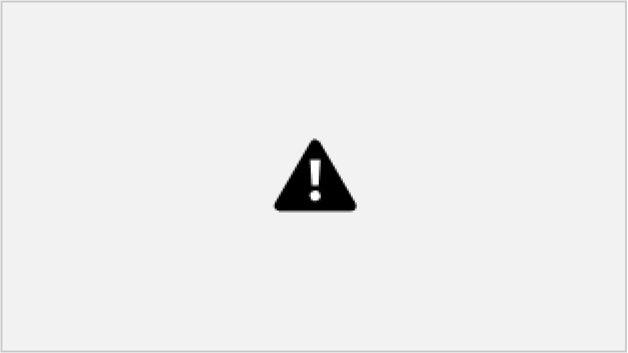
The 3D printed housing will enclose the tag reader, Raspberry Pi, and the power cord for the power supply so that they will not be affected by weather or any other external factors. This housing will be mounted on a stand outside the dismissal location so that the reader is clearly visible to both faculty and parents in case we need to move forward with passive system.

Active tags, unlike passive tags, require some kind of power source to be read by the reader. In order to solve this problem, we decided to work with cigarette lighter power adapter inside vehicles to power the tags. PCB for USB and cigarette lighter power adapter will charge the tag via USB port.

4.5.3 Screens



Active RFID tag with custom made casing/power management.



Custom made raspberry pi casing suited for outdoor use.

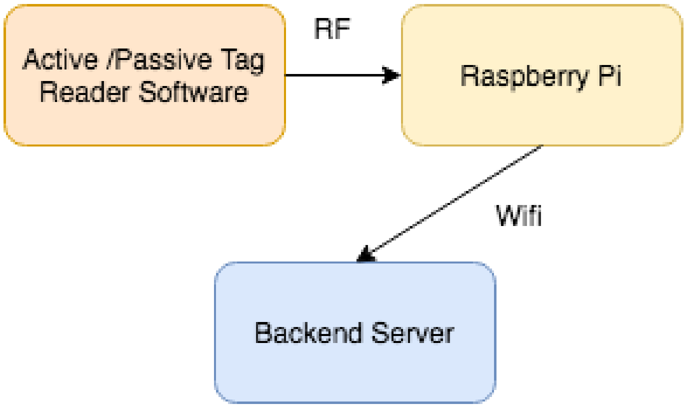
4.5.4 Experimental Evaluation

First and most important task at hand was programming the Synapse modules to be the reader and the sender. Through Portal IDE and SnapPy, we successfully programmed one module to receive the string text from the other module when in range. The string text will be an encrypted string so that backend will know from the string, which child needs to be picked up. Timestamp associated with this reading will be configured in the HTTP request creation process and sent with the text string to the backend system. The active tag's power management system through 12v cigarette jack and USB port worked successfully. We tested it on the actual car and computer.

Designs for housing were modelled and printed with the 3D printer. These housing will enclose the tag reader, Raspberry Pi, and the power cord to the tag reader. Mount for the housing will also be made at the start of the next phase. Testing between tags, reader, and Raspberry Pi will happen as soon as possible then the hardware parts will be installed onsite for full subsystem prototype testing.

Passive system with passive tags and stock tag reader successfully worked with the range of ~1m accuracy.

4.5.5 Software Architecture



The Passive or Active tag sends the pre-saved ID that are matched to the children's name to the Backend server going through, RF and Wifi data transmission. Raspberry pi in the middle supports both RF and Wifi Transmission.

4.5.6 Software Modules and Status

The Dismissal subsystem requires a software module that will receive the data from the tag reader and convert them into the formats that will work with backend group and data analytics group. This software module will be placed on the Raspberry Pi.

For active system, the string text that the reader will read from each tag can be determined by us. The tag will be programmed so that it only sends the identifiable string text to the reader. Software module will take in this string text as an input and send both the timestamp and the text as HTTP POST request and send to backend. At the end of the each day, CSV file with all the tags read that day in sequential order will be sent to the data analytics section for data analysis. For passive system, we cannot program what data will be sent and received. Each tag will have unique tag associated with it and the child. Software module for passive system will differ from that of active system since we need to link the child to the tag ID and only send the relevant information to other subsystems. Software module for this system will require a dictionary or some data structure to link the tag ID and the child to be picked up.

For active system, tag communication has been successfully tested and the Raspberry Pi has been successfully linked with CMU-SECURE wireless network. The script for active system will be tested soon with the reader, tag, and Raspberry Pi. The script for passive system will be developed by modifying minor sections of the script for active system and tested once the passive system parts arrive.

4.5.7 List of components, cost, power

Product Name	Unit Price	Quantity	Total Price	Link
Synapse Wireless SN132HO-NR	41.05	1	41.05	https://www.digikey.com/product-detail/en/synapse-wireless/SN132HO-NR/746-1039-ND/2804310
Synapse Wireless RF300PD1	42.11	2	84.22	http://www.digikey.com/products/en?keywords=RF300PD1
Synapse Wireless RF200P81	33.84	2	67.68	http://www.digikey.com/products/en?keywords=rf200p81
Omni-ID Dura 1500 RFID Tag pack of 5	39.99	1	39.99	https://www.atlasrfidstore.com/omni-id-dura-1500-rfid-tag-pack-of-5/?utm_source=google&utm_campaign=&utm_medium=cpc&utm_content=st3fYbIso_pcrid_102673901943_pdv_c&gclid=CjwKEAiA_9nFBRCsurz7y_Px8xoSJAAUqvKCgLHipXYXzb6MTaCI4onAc5TbaSdLG_BMJcP83qggAhoC0GXw_wcB
Chafon 6M Medium Range UHF Reader Writer	209.90	1	209.90	https://www.amazon.com/dp/B00NN4XUTS/ref=twister_B01G57T1WY?_encoding=UTF8&psc=1
CanaKit Raspberry Pi 3 Kit with Clear Case and 2.5A Power Supply	49.99	1	49.99	https://www.amazon.com/CanaKit-Raspberry-Clear-Power-Supply/dp/B01C6EQNNK/ref=sr_1_3?s=electronics&ie=UTF8&qid=1490926387&sr=1-3&keywords=raspberry+pi+3
URBEST®5M Vigor VH VB Series PLC Programming Cable 16.5Ft USB 2.0 to RS232 DB9	13.99	1	13.99	https://www.amazon.com/URBEST%C2%AE5M-Vigor-Programming-Cable-16-5Ft/dp/B00OZGLNW4/ref=sr_1_1?s=hi&ie=UTF8&qid=1490926525&sr=1-1&keywords=Rs232+usb
10 space 2mm header	1.35	10	13.5	https://www.digikey.com/product-detail/en/sullins-connector-solutions/NPPN121BFCN-RC/S5751-12-ND/804814
AZ1117E voltage regulator 3.3 fixed	0.43	5	2.15	https://www.digikey.com/product-detail/en/diodes-incorporated/AZ1117EH-3.3TRG1/AZ1117EH-3.3TRG1DICT-ND/5001336

10 microF ceramic cap	0.15	5	0.75	https://www.digikey.com/product-detail/en/taiyo-yuden/JMK107BJ106MA-T/587-1256-1-ND/931033
22 microF ceramic cap	0.23	5	1.15	https://www.digikey.com/product-detail/en/murata-electronics-north-america/GRM188R60J226MEA0D/490-7611-1-ND/4280544
USB mini connector	1.5	5	7.5	https://www.sparkfun.com/products/587
Cig Jack for car	2.84	5	15	https://www.amazon.com/Inkach-Accessory-Cigarette-Lighter-Connector/dp/B01MFDJ1BA/ref=pd_sbs_263_43?_encoding=UTF8&pd_rd_i=B01MFDJ1BA&pd_rd_r=QQPS40GZJ3RAB0BPMW8A&pd_rd_w=x742A&pd_rd_wg=T608v&psc=1&refRID=QQPS40GZJ3RAB0BPMW8A

4.6 Back End

4.6.1 Functionality

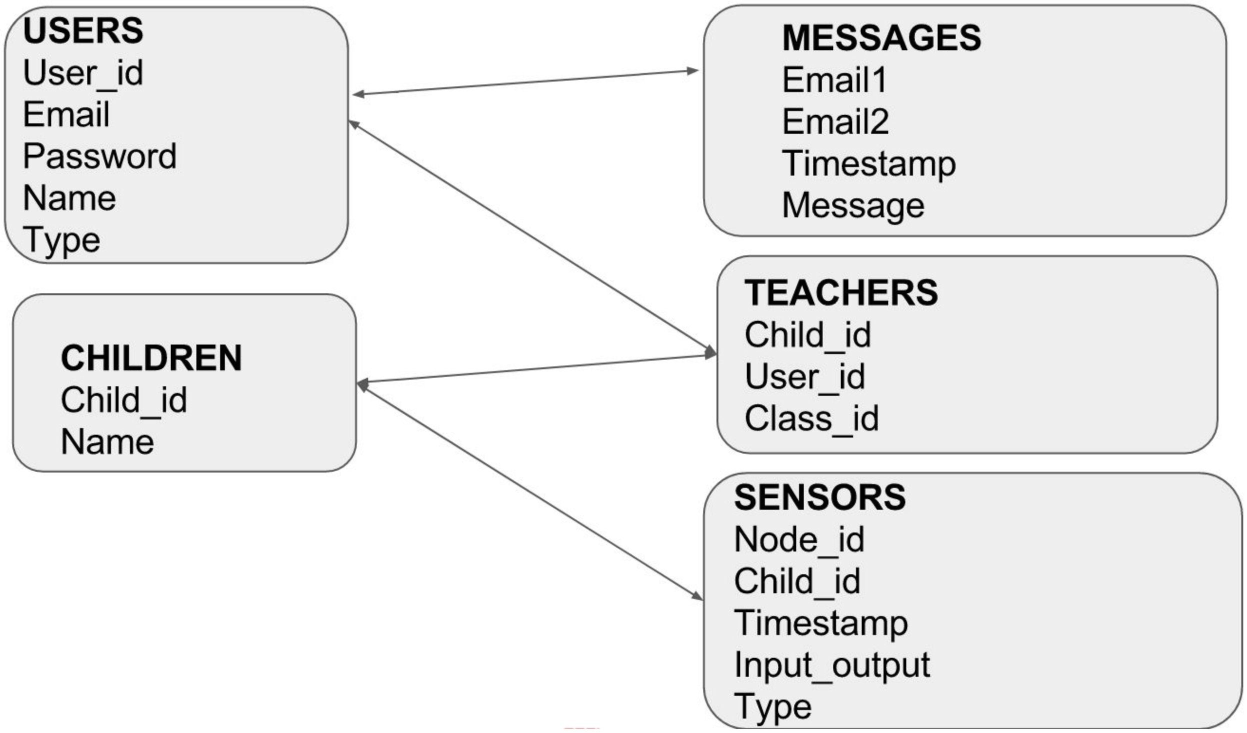
The database is responsible for mainly storing two types of data: raw data collected from the sensors, and processed data from the data analytics scripts developed by the Data Analytics team.

Storing raw data serves the purpose of preserving history data, which might become of use in the future. For example, if a teacher or a parent wishes to view an analyzation/breakdown of a specific child's activities over a specified period of time, the database supports this flexibility of fetching desired raw data entries to be processed into meaningful data. Specifically, the three types of raw data pushed into the database are data from the children's wearables, the sensor nodes placed in the classrooms, and the messages exchanged between the parents and teachers.

Another type of data stored in the database is the processed data by the Data Analytics team. This storage primarily serves as a backup so that the teachers can easily visualize children's activities on a certain day by requesting data for that day.

Currently the project has functionalities only to support data from the sensors. In future, the functionality to add schema tables for the messaging activity can be added.

4.6.2 Database and Schemas



As shown above, there are 5 major tables, which not only contain the basic information about the sensors, children, teachers, and parents, but also the mapping information. The Users tables provides information about the users that can be logged into the web application.The Children's table provides information about the students studying in the school. The sensors tables provides the data collected from the sensors.The data is taken in according to timestamp which aids data analysing. The Teacher's table provides a mapping between the sensors,users and children. Each user can either be the admin,a teacher or a parent, hence the "type" field in the "User" table. All the sensor data have a node ID, and therefore there's only one sensors table with a "type" field. This provides information as to if the data collected from the sensors tables is from activity sensing or dismissal team. Since for each of the three sensors there exists a mapping between them and the children, there is a "child_id" key for the "Sensors" table as well. For the messages table, the two email primary keys correspond to teachers' and parents' email addresses.

The illustration of the whole database tables is as below:

For example let us consider two important tables of the schema: the sensor table which stores the data from the sensors and childrens table which stores the information about the children

```
mydb3=# select * from sensors;
```

```
 node_id | child_id | type |     time_stamp
---------+----------+------+---------------------
       1 |        1 | t    | 2017-02-11 11:22:33

mydb3=# select * from children;
 child_id | name
----------+------
        1 | adam
```

The table for the sensors and children is displayed as above.

The database team will receive a .csv file from the activity sensing and dismissal teams respectively. The csv file has to be parsed and the data from the file is to be converted into required data type and then stored into the database with appropriate query requests.

As postgresql is a relational database the tables can be mapped from to each other via joins and the redundancy of data is avoided. As shown in the above two tables the child name and sensor id is not present in the same table but in order to access the child name and related sensor data we can use the following join:

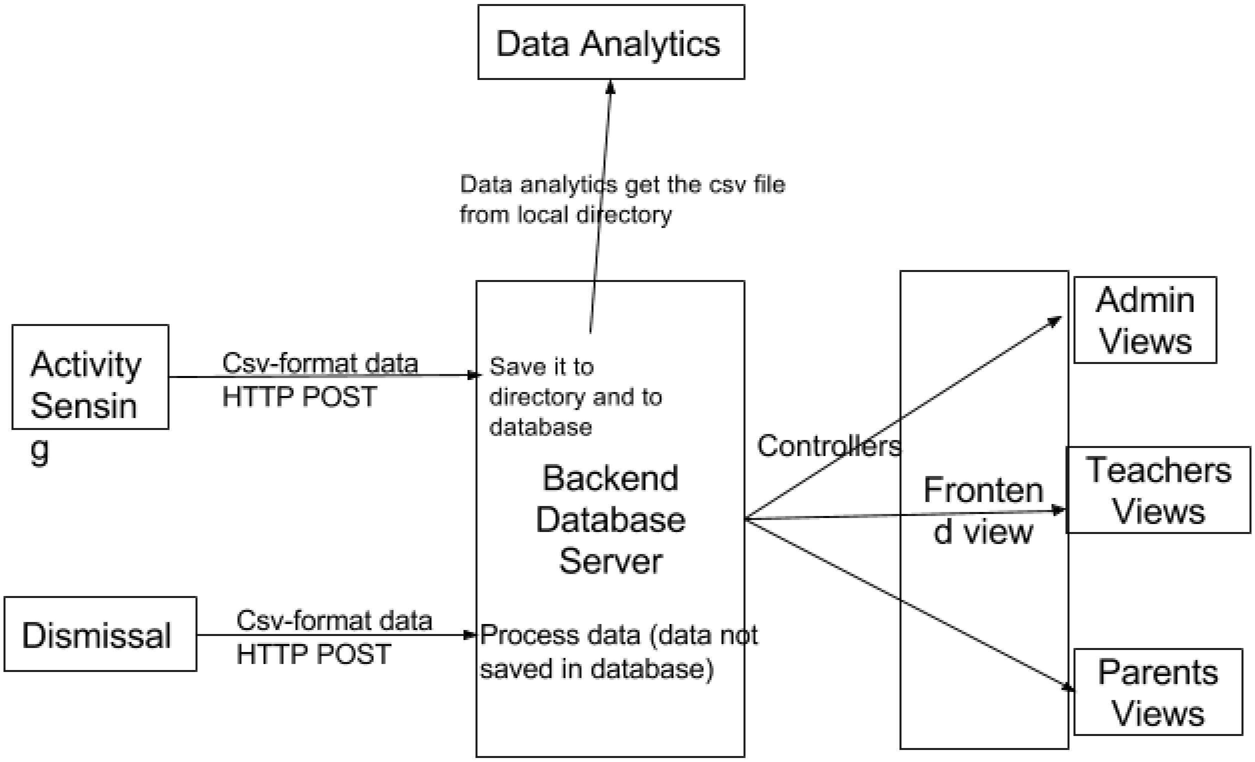
```
mydb3=# select * from children, sensors
where children.child_id = sensors.child_id;
 child_id | name | node_id | child_id | type |     time_stamp
----------+------+---------+----------+------+---------------------
        1 | adam |       1 |        1 | t    | 2017-02-11 11:22:33
(1 row)
```

In this way multiple tables can be accessed using different queries so that redundancy of data can be minimised and data can still be available on request.

Once the data has been stored in the database it has to be again provided to the data analytics team in the form of a csv file so that it can be used by them for further analysis.

4.6.3 Software Architecture

The software architecture is shown in below.


Data Analytics
Data analytics get the csv file from local directory
Activity Sensing
Csv-format data HTTP POST
Save it to directory and to database
Backend Database Server
Controllers
Fronten d view
Admin Views
Teachers Views
Parents Views
Dismissal
Csv-format data HTTP POST
Process data (data not saved in database)

As shown in figure, activity sensing sends us the data (csv format) through HTTP POST. After receiving the data, the backend team runs the bash script, which save the data in the local directory and database. The reason for saving it to the local directory is because the data analytics team need to retrieve the data from local directory. Dismissal team would also sends us the data (csv format) through HTTP POST. However, due to its real-time functionality, there is no need for saving dismissal data into database. As a result, for dismissal data, after receiving it, it would be parsed and sent to front-end directly.

All the data inside database server would be sent to front-end based on different views upon requests from front-end. Controllers have been developed based on different views.

4.6.4 Experimental Measurements

We tested the communication between activity sensing and backend server, as well as between dismissal and backend server. JavaScript files were developed to achieve this functionality. The javascript files can be accessed from the application code.

On testing the dismissal controller at the server with the dismissal RFID, the tags corresponding to the children’s name were detected in order and printed on the screen in order. The child id corresponding to the tag id was then fetched by the front end in the order of detection.

We also tested the controller's’ accuracy. First we populated mock data into database, shown in following.

```
rpcsdb=# SELECT * FROM users;
 user_id |    email    | password | name |  type
---------+-------------+----------+------+---------
       1 | 1@gmail.com | asdf     | Alex | admin
       2 | 2@gmail.com | awfd     | Lily | teacher
       3 | 3@gmail.com | wojd     | Noe  | teacher
(3 rows)
```

```
rpcsdb=# SELECT * from teachers;
 user_id | child_id | class_id | name | age | allergies |  birthday
---------+----------+----------+------+-----+-----------+------------
       2 |        1 |        1 | Lily |  50 | shellfish | 01/01/1967
       2 |        2 |        1 | Lily |  50 | shellfish | 01/01/1967
       2 |        3 |        1 | Lily |  50 | shellfish | 01/01/1967
       3 |        1 |        1 | Noe  |  50 | shellfish | 02/01/1967
       3 |        2 |        1 | Noe  |  50 | shellfish | 02/01/1967
       3 |        3 |        1 | Noe  |  50 | shellfish | 02/01/1967
(6 rows)
```

```
rpcsdb=# SELECT * FROM children;
 child_id | name | age | allergies |  birthday
----------+------+-----+-----------+------------
        1 | John |   3 | shellfish | 09/01/2014
        2 | Lisa |   4 | shellfish | 09/23/2013
        3 | Tom  |   5 | shellfish | 09/01/2012
(3 rows)
```

Then we tested the queries to make sure we got the accurate results. Followings are the test results for admin controllers.

QUERY 1 – Admin Login (selecting and comparing password)

```
rpcsdb=# SELECT password FROM users WHERE email='1@gmail.com';
 password
----------
 asdf
(1 row)
```

As you can see we got the correct password.

QUERY 2 – Admin view (a list of teachers)

```
rpcsdb=# select *
rpcsdb-# from (
rpcsdb(#    select *,
rpcsdb(#           row_number() over (partition by user_id) as row_number
rpcsdb(#    from teachers
rpcsdb(#    ) as rows
rpcsdb-# where row_number = 1
rpcsdb-# ;
 user_id | child_id | class_id | name | age | allergies |  birthday  | row_number
---------+----------+----------+------+-----+-----------+------------+------------
       2 |        1 |        1 | Lily |  50 | shellfish | 01/01/1967 |          1
       3 |        1 |        1 | Noe  |  50 | shellfish | 02/01/1967 |          1
           (2 rows)
```

As you can see, we got a list of teachers successfully.

QUERY 3 – Admin view (a list of children)

```
rpcsdb=# SELECT * from children;
 child_id | name | age | allergies |  birthday
----------+------+-----+-----------+------------
        1 | John |   3 | shellfish | 09/01/2014
        2 | Lisa |   4 | shellfish | 09/23/2013
        3 | Tom  |   5 | shellfish | 09/01/2012
(3 rows)
```

As you can see, we got a list of children successfully.

QUERY 3 – Admin view (a list admin)

```
rpcsdb=# SELECT * from users WHERE type='admin';
 user_id |    email    | password | name | type
---------+-------------+----------+------+-------
       1 | 1@gmail.com | asdf     | Alex | admin
(1 row)
```

As you can see, we got a list of admins successfully.

QUERY 4 – Admin view (see a particular teacher)

```
rpcsdb=# SELECT DISTINCT user_id,class_id,name,age,allergies,birthday from teachers WHERE name='Lily';
 user_id | class_id | name | age | allergies |  birthday
---------+----------+------+-----+-----------+------------
       2 |        1 | Lily |  50 | shellfish | 01/01/1967
(1 row)
```

As you can see, we got the correct teacher.

QUERY 5 – Admin view (see a particular child)

```
rpcsdb=# SELECT * from children where name='John';
 child_id | name | age | allergies |  birthday
----------+------+-----+-----------+------------
        1 | John |   3 | shellfish | 09/01/2014
(1 row)
```

As you can see, we got the correct child.

Query 6 – Admin view (see a particular admin)

```
rpcsdb=# SELECT * from users WHERE type='admin' and name='Alex';
 user_id |    email    | password | name | type
```

```
---------+-------------+----------+------+-------
       1 | 1@gmail.com | asdf     | Alex | admin
(1 row)
```

As you can see, we got the correct admin.

4.6.5 Software Modules and Status

The three tasks of the project form three modules.

Users Module

The frontend requires information about users which includes parents, teachers and admin. Hence, routers in Node.js for each one of them are used to access profile specific information such as allergies, birthday etc. This information is entered into the database when a user signs up for the service and fetched whenever the user logs in.

Dismissal Module

The dismissal RFID server sends the identification number on detecting the arrival of a guardian for pickup. The id is received by the Node.js controller and used to query the database and retrieve the child id corresponding to it. The child id is then added to the dismissal table in the database. From there, the frontend periodically access all the newly added child ids and updates the view for teachers and admin.

Activity Sensing Module

The activity sensing data is retrieved in a csv file using a predefined format and entered into the database using a bash script. The csv file is also used by data analytics to create visualizations for the frontend. The sensors’ data stored in the database can then be used by the frontend by calling routers to fetch sensors' information.

4.7 Conclusions

4.7.1 Requirement Features Table

		Requirement Fulfilled By:				
Category	Task	Frontend	Data Analytics	Activity Sensing	Dismissal	Backend
Dismissal	Display Interface	X			X	
Communicati on	Teacher Interface s	X	X			X
Student Activity Tracking	Teacher Interface	X	X	X		X
	Parent Interface	X	X	X		X

Table 4.7.1 Table depicting which teams have taken on each key requirement

5. Project Management

5. Project Management

5.1 Time Logs and Responsibilities

5.1.1 Front End

5.1.1.1 Work Logs

Team members	Class	Interface Design	Research	Preparing Spcs	Group meetings	Administrative Tasks	Coding
Sarah Klein	4 hours	6 hours, 55 min	--	--	45 min	6 hours, 45 min	--
Kavya Kumar	14.2 hours		3.5 hours	15 hours	6.75 hours	6.5 hours	--
Leann Bahi	11.5 hr	17.75hr	3.5 hr	--	--	10.5 hr	--
Abhishek Jindal							
Luke Miller	4h					2h	10h
Jack Gao	12h	1h	2h	12h	2h	1h	2h
Sujay Kotwal	12h	--	3h	--	3h	2h	--

5.1.1.3 Summary by Group of Individual Student Contribution

	Team members	Roles	Each member's contribution	Team contribution	Team hours
Front End	Sarah Klein	Click-through demo creation Diagram creation Dismissal interface creation	Creation of parent mobile and parent messaging click through demo Creation of final versions of system architecture and interaction flow diagrams Design of dismissal interface		
	Kavya Kumar	Leader, Demo Script Head, Demo integration Liaison	I was the leader of the Phase 3, so I oversaw the functioning of the different subsystems of the frontend I've been involved in communicating tasks back and forth between the visualization team and my front end so that the web app could show the relevant information I contributed in writing the demo script and was in cahoots with the demo master. I also set up the dry run and managed the presentation of the actual demo of the project		

	Leann Bahi		Finished high fidelity wireframes for teacher and admin interfaces. Updated all high fidelity wireframes with more realistic data (names, text, activities...) Created click through demo using invision for teacher, admin, and teacher messaging interface (backup for final demo).		
	Abhishek Jindal				
	Luke Miller	Frontend programming	Luke wrote CSS for the parent and dismissal views. Luke also worked on the final presentation slides, particularly the introduction and parent view slides.		
	Jack Gao	Editor	Compiled the front end sections of the phase 3 report. Contributed to the demo script. Prepared demo stage map and demo flow. Played the teacher role in the final demo.		

	Sujay Kotwal		Acted as a liaison between dismissal and front end team. Contributed to demo script. Formulated character assignment and demo setup.		

Team members	Class	Interface Design	Research	Preparing Spcs	Group meetings	Administrative Tasks	Coding	User Testing	Field Testing	Leadership Meetings
Sarah Klein	36 hours	95 hours, 45 min	30 min	--	45 min	11 hours, 50 min	--	--	15 min	--
Kavya Kumar	40 hours	10 hours	12 hours	28 hours	26.5 hours	7 hours	3.15 hours	--	60 min	--
Leann Bahi	43.5hr	34.25hr	5hr	0	17.75hr	16.5hr	0	2hr	45min	5hr
Abhishek Jindal										
Luke Miller	48h	20h	10h	0	38h	10h	10h	6h	0	5h
Jack Gao	42h	10h	20h	15h	12h	5h	5h	2h	0	0
Sujay Kotwal	43h	15h	22	5h	15h	16h	0	1h	30mins	0

5.1.2 Data Analytics

Tasks and Assignments:

Data Analytics:

Code for plotting functions and implementation of the pickle module: Sam
Research and code for further plotting functions, file I/O, interactions with frontend application (Javascript + Python subprocess), matplotlib/plotly: Kevin
Code for CSV file parser, mock data and integration of plotly: Karim
Optimizing output histogram: Sheryl
Research and compare different python modules and implement pie charts using plotly: Radhika
Compare matplotlib and plotly and select the best module to work with: Vanshaj
Research data visualization libraries: Abhishek

Summary of Work Log Hours:

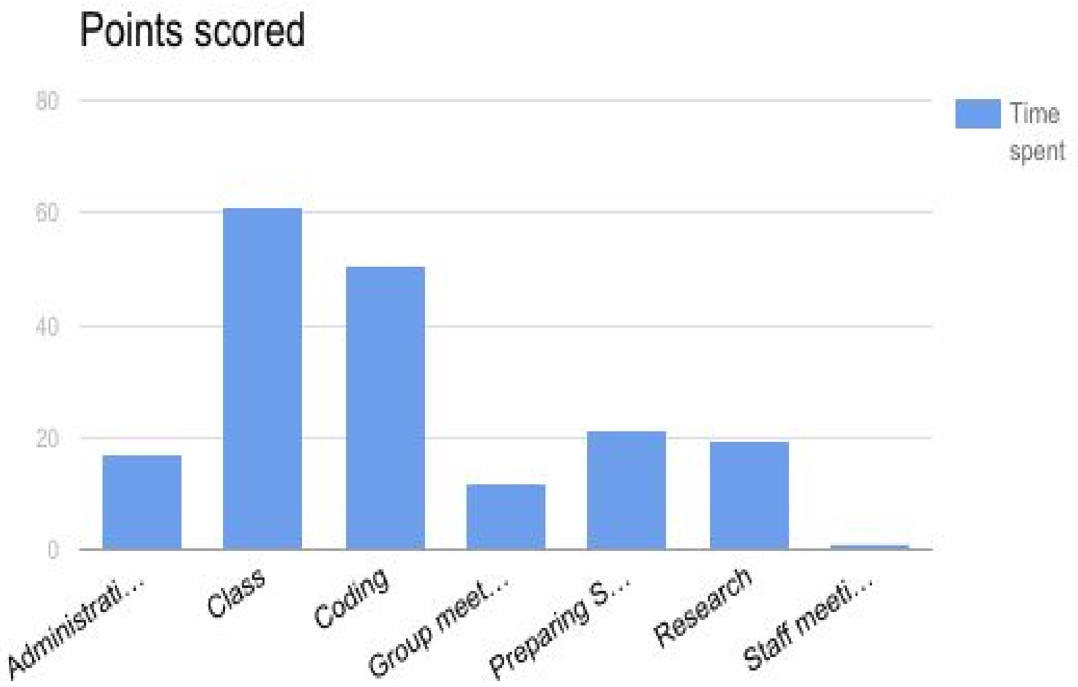
	Team Members	**Roles**	**Each member's contribution**	**Team contribution**	**Team hours**
Data Analytics	Karim Elmaaroufi	Leader	Karim worked on the CSV parser and on integrating the plotly module into our code. He also wrote the mock data to test our code	As a team, we worked on writing the code for all our functionalities, test and debug our code and generate graphs that depict all our functionalities pictorially. These graphs were then sent to the frontend team as png files so that they can display	220 hours
	Sam Xu	Liaison	Sam worked on writing the code for three plotting functions. He also used the pickle module to write code that keeps records for future use		

			during data queries.	them on the dashboards	
	Sheryl Zhang	Presenter	Sheryl worked on optimizing the output histogram and on the presentation for phase 3		
	Vanshaj Sikri		Vanshaj researched and compared matplotlib and plotly, edited the script to add changes suggested by the reviewers and also helped Sheryl with the presentation		

	Radhika Amare	Co-editor	Radhika researched different graph modules like plotly, networkX and iGraph and decided to use plotly as the final choice. She worked on debugging and testing the code with the mock data. She also worked on the Phase 3 report as she was one of the two editors. She was also part of the demo		
	Kevin Chon	Co-editor, Liason	Kevin implemented further data analyses including average time spent per visit per student, median time per visit for all students per station, all visits per station, etc. Kevin also researched into how to properly access plotly (use plotly offline) as well as both Python		

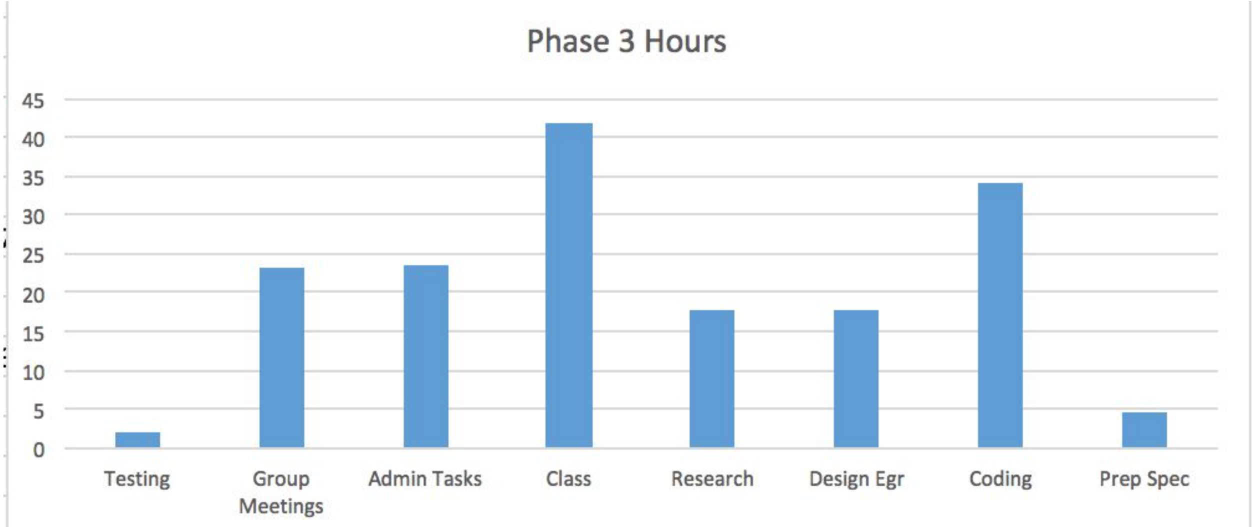
			and Plotly file I/O capabilities. Futhermore, Kevin researched and coded with the front-end team in Javascript to determine communication to our script.		
	Abhishek Yadav		Researched data visualization libraries for better rendering		

Histogram of time by topic



5.1.3 Activity Sensing Project Management

5.1.3.1 Work Log Hours Histogram

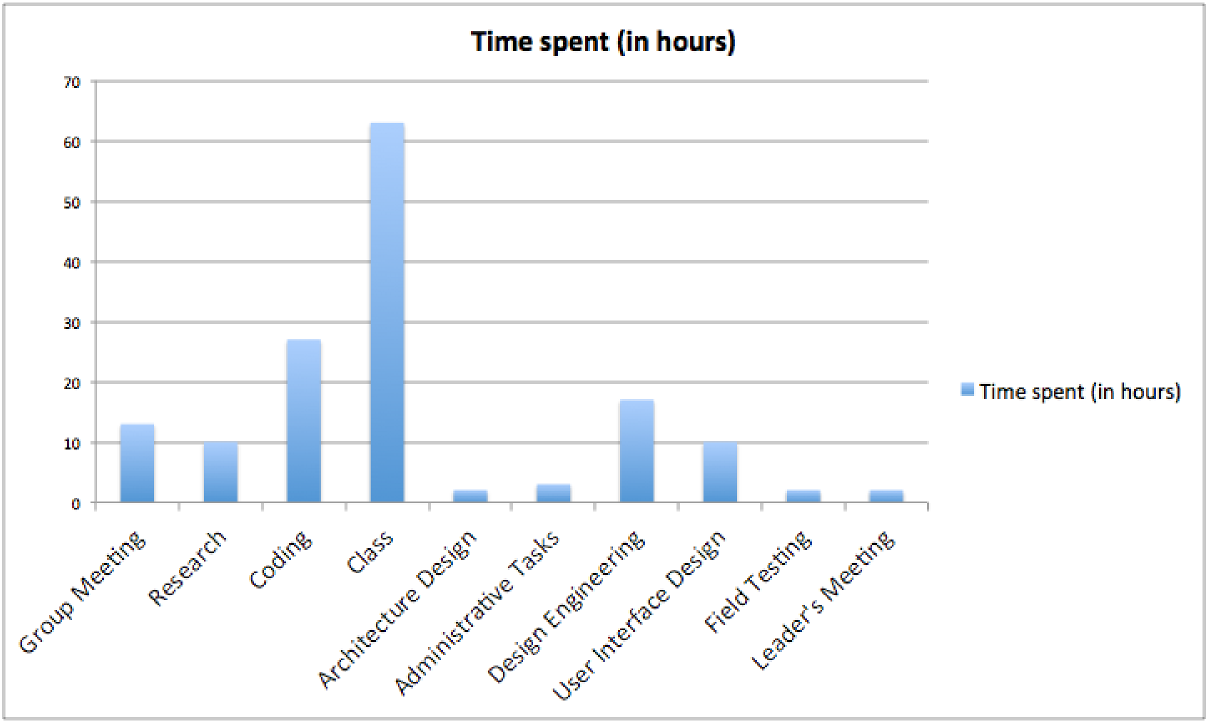


5.1.4 Dismissal Project Management

5.1.4.1 Summary of Work Log Hours for Phase 3 By Category

Topic	Time spent (in hours)
Group Meeting	13
Research	10
Coding	27
Class	63
Architecture Design	2
Administrative Tasks	3
Design Engineering	17
User Interface Design	10
Field Testing	2

Leader's Meeting	2



5.4.3 Summary by Group of Individual Student Contribution

Member Name	Role	Individual Contribution
Arani Basu	Leader	Held team meetings, coordinated with other teams for the demo, helped set up the pis for active/passive demonstration
Sujay Kotwal	Presenter	Presented the Phase 3 demo, modified encasement of raspberry pi, helped demo the whole system
Kyuin Lee	Editor	Modified active tags code, setup raspberry pi's dependencies, developed PCB and modified it, Edited phase 3report
Akshit Sharma	Member	Modified active tags code, tested the whole system several times with active/passive tags,

		Actively worked on active system code
Kyle O’Rourke	Member	Designed and manufactured active RFID encasement, Demoed the final demo, helped set up for the demo
Jennifer Meagan Lloyd	Member	Setup all dependencies on raspberry pi side, configured raspberry pi, worked some more on the passive tags, modified code for passive/active tags
Sally Lee	Member	Testing active reader, communicated with backend group for database communication, took care of administrative tasks
Amanda Marano	Member	Coordinated demo masters, held several out of classroom meetings, communicated with different teams to finalize everything

5.1.5 Backend Project Management

Backend Database Schema Design: All team members

Admin View Controllers: Karen Zhou

Teacher View Controllers: Gayatri Kamat

Parents View Controllers: Momina Haider

Contacting Points to Data Analytics: Karen Zhou and Momina Haider

Contacting Points to Activity Sensing: Gayatri Kamat

Contacting Points to Dismissal: Momina Haider

Contacting Points to Frontend: Abhishek Jindal

5.2 Lessons Learned

Here are some lessons learned submitted by class members.

Big teams need a lot of managing
Teamwork. Distribution of work is very important to achieve all objectives in a timely manner
If they can get away with it, a person will do nothing until a job is taken from them

Interfaces need to be very carefully considered before moving forward

Diversity of ideas is valuable

5.3 Suggestions for Improving Class

First, let's go over some things that were done well.

The professors were always willing to give us work from prior classes, to use as examples. However, it would have been helpful to be more targeted in how to go about finding the right information from those. (HLW: Mastery: Explore available educational materials).

The professors made it very clear what was to be rewarded for each phase. Phase 1 was communication on Kiva, and Phases 2 and 3 were "owning something." (HLW: Motivation Establish Value: Identify and Reward what you value).

Dan's cheery disposition and breadth of experience of the subject was often encouraging. (HLW: Motivation: Identify and Reward what you value).

Dan was consistently tying course goals back to higher-level skills like communication, planning, and project management. (HLW: Motivation: Demonstrate the Relevance of Higher-Level Skills to Students' Future Professional Lives).

Now, what could be improved

There was practically no assessment of prior knowledge. How would we know who had engineering skills? Who had design skills? Who had project management and leadership skills? Many of us found ourselves in a classroom full of people of whom we had no idea what they were studying, who they were, and what they were interested in. If there were more team building activities, there could be greater class cohesion, and thus greater motivation. Here are some potential solutions:

- Providing some sort of "icebreaker" activities. Perhaps have students prepare five to ten minute presentations on themselves to present at the individual group meetings, in order to express their goals, their experience, their expectations, what they're excited about, etc. This could help increase team building and motivation.

The class followed a "waterfall" development strategy, i.e. one big project that would either succeed or fail. Perhaps starting smaller with a minimum viable product that can be iterated on would be better? This aligns with the strategy in *How Learning Works* to "provide early success opportunities." In class, it seems we didn't have the chance to build anything until the very last phase.

To increase motivation, the visionary scenario should have been more… *visionary*. Instead of just describing what goes on in one classroom, students should express the *big picture*. The CMU Children's school is at the forefront of educational research, and let's bring it to the forefront of technology! We can build RFID readers, sure, but what are the implications for the future? How else can we instrument classrooms? Sure, it helps Mrs. Darsh and whatnot, but why is that important? It maybe provides more convenience, but as Karen Bernstein teaches in IXDO, "you'll never get anyone excited about convenience." After we build this product, what would other potential next steps be? Maybe electronic, sensored toys at every station, eye-tracking software, and etc. If we can get students thinking big, maybe we could have more students "owning something".

Some suggestions from classmates:

Make Kiva better - switch to a mobile version, have some better way to fill worklogs.
Indeed, there were many complaints about Kiva. It can be especially troublesome when teams are already using Slack to communicate and Google Drive to share folders. It is 95% likely that you could use Slack to achieve all of the functionality of Kiva. You might have to get a little creative by writing some plugins, but it can probably be achieved. Also, if you use Slack, students can possibly write *their own* plugins to help with communication, and it will give them a greater sense of control over the communication process.

It's challenging to coordinate an 8 person team to meet, and it's hard to split up work enough so that everyone can operate independently. Smaller teams would arguable fix both of these issues.
Having to make everything into one system was a limiting factor for a class of this size. Perhaps with forty to fifty students, having two separate teams compete would be better. This way, there are smaller team sizes, and there's an incentive to compete and perform.

Have more accountability up front; create more leadership positions for previously unaccounted for activities so they need to get done.

Faster iteration cycles (or phases within phases).

A flow chart of how the complete system will work should be prepared before proceeding to code.