University of Arkansas – CSCE Department
Capstone I – Preliminary Proposal – Fall 2022

GateMate

Jackson Bullard, Nathaniel Fredricks, Jose Martinez, Carissa Patton, Ivris Raymond

Abstract

The use of alternate wetting and drying (AWD) is gaining popularity among rice farmers in Arkansas as a way to reduce water usage without sacrificing crop yield. Doing this effectively requires frequent adjustment of levee gates, and timely accuracy is necessary in order to avoid diminished crop growth. This is difficult to achieve with traditional gates, where workers must manually adjust each gate, a process that is both time consuming and prone to human error. Gates remotely controlled by a stakeholder through software such as a mobile app would provide an effective solution to this problem.

The GateMate software will be developed to utilize and interface with an existing GateMate device prototype, along with the sensor array present on the prototype and any necessary field tracking technologies. The GateMate software application will not only allow rice farmers to raise and lower gates in the field remotely, but it will also factor in weather conditions to automatically adjust the gate levels to conserve the most water as well as assisting with optimal initial placements of the gates.

The architecture of the GateMate application will consist of three separate pieces of software: the mobile application that the users interface with, the central server and database, and the gate software that will run on each gate's control circuit. The mobile application will be implemented using React Native, which is a JavaScript framework used for writing natively rendering mobile applications for iOS and Android. A database will be used to store the necessary geographical and weather data needed for optimal gate placement. Finally, the third piece of software will network the gates using the ESP8266 Wi-Fi Mesh API, allowing the user to issue commands to individual gates or groups of gates through the React Native Application.

The significance of an application like this involves resolving many issues faced by rice farmers today, including aiding in water conservation. Being able to remotely control gates will help farmers reduce excess water consumption, allowing farmers to save money and use optimal water levels to maximize crop yield. Additionally, one with less irrigation experience would be able to use the GateMate software to identify the most advantageous gate placements on a given topography as well as automate the process of modifying gate heights to regulate water levels on the field.

1.0 Problem

Rice, wheat, and maize provide over 50% of all calories consumed by the world population [1]. Alone, "rice provides 21% of global human per capita energy and 15% of per capita protein" [1]. Arkansas farmers harvest over 200 million bushels of rice on an average of 1.3 million acres annually, and, in 2015, Arkansas accounted for 49% of the United States' total rice production [2]. More recently, rice production in the U.S. exceeded $3 billion in 2021 [3].

While many irrigation techniques exist, an increasingly popular technique referred to as alternate wetting and drying (AWD), also known as intermittent flooding, is gaining traction in Arkansas [4]. Already popular in surrounding states, this water management practice is intended to reduce water usage and greenhouse gas emissions without sacrificing crop yield; however, it can be labor intensive due to the need to frequently adjust gate height in order to precisely control water level. The current stage of growth of the crop, soil characteristics, and weather conditions can all affect the desired level of water. Furthermore, with so many factors in play, there are many chances for human error year after year.

In order to regulate traditional gates to maintain desired water levels, workers must travel out into the (usually flooded) field and manually adjust the gates. This is time consuming and error prone. Workers may not always appropriately adjust the gates, and farmers may incorrectly deduce the proper gate height; many gates can feed into the source of others, creating a complicated irrigation chain that complicates the decision-making process. Furthermore, workers may neglect adjusting the gates entirely, and the probability of this increases with the difficulty in reaching a gate and the granularity with which the stakeholder wishes to control the water levels. Precise control can be necessary for maximizing crop yield, and lost yield is a cost intolerable to the modern farmer.

As the world's population grows, the Earth's natural resources bear increasing strain. Some estimates suggest that the near-to-mid future may require as much as 40% more water than is currently accessible [5]. Furthermore, water withdrawal by rice can be significantly higher compared to other field crops (depending on method of irrigation) [6]. AWD is designed to minimize wasted water, and automated, weather-informed irrigation systems present an opportunity to utilize rainwater to further reduce the need for water withdrawal. In addition to increased water conservation, every acre-inch of groundwater that is offset by rainwater capture (or otherwise not pumped from a surface/groundwater source) could save as much as a gallon of diesel fuel [7]. The opportunity to conserve natural resources without sacrificing yield necessitates investigation. A solution would benefit both rice farmers' livelihoods and the Earth's increasingly strained ecosystems.

2.0 Objective

- Required Objectives
  - Interface to raise and lower gates
  - Application to assist with the initial placement of gates
  - Application automatically raises and lowers gates according to…
    - Weather (rainfall)
- Optional Objectives (Stretch Goals)
  - Application automatically raises and lowers gates according to…
    - Crop height
    - Crop growth rates

A feature complete implementation of a solution should provide farmers an interface for the GateMate that is available at least on their mobile phones and ideally across several platforms. The GateMate application should provide most of this information through a map interface that displays the user's field and gates.

The minimum viable product version of the GateMate software should allow farmers to raise and lower the gates manually, taking into account the topology of the land on which the gate array sits. The software should also assist farmers in optimally placing their gates given their field's topographical information. This is important as the gates will differ in elevation, so allowing 4 inches of water into one section of a field may mean raising a gate more or less than another gate controlling the water level in another section of the field. This degree of implementation will at least accomplish the objective of making it possible to move the gate positions without having to be physically present to do so. Finally, at a minimum, the gates should consider weather conditions when automatically making adjustments to elevation. Currently, a farmer must carefully track the weather and rainfall and take that into account for their calculations for the appropriate setting of the gate elevations. Ideally, the GateMate software would be able to do this for the farmer. A simpler version of this that likely would not require much additional work would allow the farmer to input the amount of predicted rainfall. Software would then automatically adjust all of the gates at once based on that amount. However, it would be much more desirable to enable intelligent decision-making in software using information supplied by some weather API, such as the Google Earth Engine API [8]. The system could then use that information to automatically adjust gate heights at the appropriate time. This feature would prevent accidents that occur from time to time in which the farmer neglects to adjust for rainfall, resulting in reduced crop yield.

Finally, our project should allow the farmer to instead input just the current height of the crop, then use that information to automatically calculate the amount of water the crop needs. The primary variable concerning a rice crop's water level requirement is the current height of the crop. The GateMate software should receive periodic crop height measurements (input by the user), likely in various locations throughout the field, then automatically calculate the amount of water required and make adjustments to the gate elevations based on that calculation and the amount of water available in the system. Furthermore, it would be possible to have the user input the date the crop was planted. The app could then prescribe water flow based on that and a model for the expected growth of the plant based on other factors such as the amount of sun the crop is getting. Since the application would already take weather into account, constructing this model would likely not be beyond the GateMate software, but the accuracy of such a model is yet unknown, so periodic height measurements done by the user would likely still be necessary.

An application that implements the above outlined features would certainly meet the objectives for this project, as the primary goal of the GateMate project is to make it easier for farmers to manage the water usage of their crops throughout the growing season. Implementing even just the minimum viable product version of the GateMate software would mean that a farmer does not need to physically raise and lower every gate to make adjustments and could do so from their phone. To also account for weather and to further automate the water adjustments based on just the height of the crops would further reduce the workload for farmers and potentially result in greater yields due to fewer errors in water management. This technology will enable fewer farmers to manage larger fields, thus making it easier to keep up with the ever-growing demand for food as the world's population continues to grow.

3.0 Background

3.1 Key Concepts and Technologies

There are many key concepts and technologies related to this problem, including the concept of alternate wetting and drying as well as technologies such as the MSP430 microcontroller, ESP8266 Wi-Fi module, and more. While it is important to understand what each key concept and technology entails and is capable of, it is even more important to consider how each of these falls into place in the larger picture of this project.

As the proposed software system will be used in agriculture to regulate water levels on fields, it is critical to understand the current utilized method of alternate wetting and drying (AWD), which is a form of watering fields. The "drying" part of AWD is when the flood is allowed to recede, typically until the soil surface is visible [5]. Once the water level reaches its intended low point, the fields are flooded again (the "wetting" part). This alternation is performed instead of maintaining a continuous flood in an effort to reduce the amount of water used. One way of implementing AWD is by controlling water flow by raising and lowering the rice paddy's levee gates, and the software proposed here will help make controlling an entire field of gates far less time consuming and labor intensive.

In past semesters, students have worked on other aspects of the GateMate device, with the current gate design containing a box with computer hardware to control the gate. The circuit consists of two custom printed circuit board (PCB) designs. One PCB contains the main processor in the circuit: a MSP430 microcontroller, which is the "brain" that makes calculations and controls peripheral devices. An ESP8266 Wi-Fi module is attached to this PCB, which connects the module to the microcontroller. The direct current (DC) motor is also connected to this PCB. The second PCB is responsible for recharging the battery using the attached solar panel. The PCB with the MSP430 can be programmed to run software that is designed for it. This software can be loaded onto the board from a computer using Energia IDE, which is a development environment for microprocessors and microcontrollers.

A mesh Wi-Fi network is a group of devices that act as a single network. Instead of a centralized source of Wi-Fi, such as a router, each device in the group essentially acts as a node, with each node offering Wi-Fi access to the network [9]. For example, the ESP8266 Wi-Fi module attached to the gate's controller could be set up to connect with other nearby gates and create a network that could then be accessed by connecting to any one of the gates.

An application programming interface (API) is an interface that allows developers to write software that interacts with another piece of software [10]. Whether it is an application, service, or some type of data, an API can make meaningful interactions easier. The ESP8266 API contains libraries for configuring the Wi-Fi module on the gates. Instead of writing and reading abstruse values to the memory module, the API offers abstracted functions that simplify the process [11]. This will be a great tool when creating a more complex system such as a mesh Wi-Fi network.

React Native is a JavaScript library used to build user interfaces [12]. When it comes to mobile phones, the two main operating systems are IOS and Android. Despite both operating systems running on mobile devices, developing for either platform is usually significantly different from the other. However, React Native is cross-platform, so it can compile to work on both IOS and Android, allowing developers to create one app that can then run on multiple platforms.

A database is a "collection of information" that is usually designed to provide efficient retrieval of specific data [13]. When it comes to servers and software, databases are often implemented using standardized technologies such as Structured Query Language (SQL), which is utilized in many databases as a common means to query, add, remove, and change data in a database [14].

In summary, this project will utilize various technologies including React Native, databases, and mesh Wi-Fi networks. All of these technologies will come together in the GateMate project to allow farmers to adjust their gates remotely.

3.2 Related Work

Over time, technology has played a crucial role in rice farming. Impacting both sustainability and profitability, farmers have been looking to technology to improve their methods and reduce costs [15]. In the past 20 years, farmers have reduced water usage by 53% and land usage by 35%, partly due to the shift from the constant flooding of fields to intermittent irrigation [15]. This newer approach to irrigation has even more room for improvement through further technological development, reducing wasted water while still allowing the fields to produce the highest yields possible.

Researchers have developed automated systems similar to this GateMate project, but they fall short of being ideal solutions. One software system, Delta Plastics' *Pipe Planner*, allows users to manually draw fields on a map interface and place gates on the fields wherever the user selects [16]. Firstly, *Pipe Planner* displays the farms previously added by the user along with the option of creating a new field. When the user prompts the application to add a new field, a map is displayed to the user that allows them to enter the location of their desired field whether that be a furrow or levee. The user is given the option to either draw a field or upload a file to automatically generate field boundaries. It also allows the user to specify a water source within the field view. However, while this application allows users to note locations of levees, it does not tell users the ideal place to position gates to regulate the water levels, nor does it provide automatic raising or lowering of these gates to ensure appropriate water levels. The GateMate will not only allow users to select their field but also will provide information as to the best locations to place gates on those fields and automatically modify the height of the gates based on water levels, the weather forecast, and conditions for best growth. While the *Pipe Planner* has features that will not be implemented on the GateMate software such as setting flow rates of nearby furrows, the GateMate will provide predictive information to place gates at locations to optimize water usage and allow farmers to either utilize this information regarding raising and lowering the gates or manually change the height of the gates, two key features that are not provided by the Delta Plastics' *Pipe Planner*.

Other systems have been created that interface with devices out in the field similar to the GateMate design. A Mississippi State University irrigation specialist, Drew Gholson, utilized ultrasonic water level sensors, actuated values, and a pump controller to attempt a proof of concept in automating the process of irrigating fields [17]. His system allowed users to "remotely monitor water levels" and turn wells on or off from a smartphone or computer so that farmers would not have to drive out to specific fields to turn off wells when a certain threshold was reached [17]. This system was based solely on a 4-inch flood threshold level, but it would automatically turn off wells when that level was reached so that excess water would not be wasted. It would also allow farmers to check water levels and well status remotely. In testing, the system averaged 223 bushels per acre as opposed to 219 bushels per acre for farmer irrigated fields, and, in terms of water used, the automated fields averaged 18.8 acre-inches compared to the farmer-irrigated fields, which averaged 28.6 acre-inches [17]. While crop yield was not significantly increased, excess water usage was reduced, thus saving farmers money and time from not having to manually turn off the wells. While effective, this system still relied heavily on the farmer to make modifications based on incoming weather, their different topology, or desired threshold levels; GateMate, however, will account for weather, topology, and optimal AWD practices to almost entirely automate the process so that minimal effort is required for micromanaging field levels. The farmer would still have the capability to use their smartphone or computer to check and maintain water levels, but the system for checking and holding water levels will be much more dynamic than the Mississippi State product. The gate in this project would allow for precise modifications of how much water is being let into the field that would be automatically set based on various conditions as opposed to the farmer having options of only turning on or off the well that is allowing water to flow. Overall, the system to be designed will account for much more data, allowing the system to be truly automated, and have greater potential to save water and labor compared to the system designed by Gholson.

Another company, MimosaTEK, which is a Vietnamese start-up, developed technology that can be used to address some challenges that are faced by farmers when they attempt to implement alternate wetting and drying (AWD). With a smartphone, a farmer can monitor the actual and recommended water levels, which allows them to "determine the best time to irrigate the rice and the optimal amount of water to apply" [18]. Water pumps can be operated by this smartphone application with mobile or internet connectivity and electricity. This system runs "reliably with an uptime close to 100 percent," indicating that it would be likely to be trusted by farmers if they can accurately and precisely see water levels on their fields at any given time. Their results showed that utilizing IoT allowed for 13-20% reduced water usage compared to farmers using conventional AWD [18]. This system utilizes a smartphone and a cloud-based management software, which will be similar to the GateMate software. An electrical engineering team has already installed hardware that utilizes Wi-Fi for connectivity, so this can be used to get information on water levels and modify the amount of water that is being let into the fields by the gates. However, the MimosaTEK system can only turn a water pump on or off rather than slightly modifying gates to allow certain amounts of water to flow to individual pools within the field, which is a key characteristic for the proposed GateMate device.

RiceAdvice, a mobile app by AfricaRice, provides "farm-specific advice on rice management practices" [19]. Guidelines are generated for farmers based off of a short interview, and it provides services such as identifying the best choice of fertilizer, determining when and how much fertilizer should be used, and allowing farmers to set their target yield levels. With the "personalized and profitable recommendations," users have increased paddy yield by 0.5-1 tons/hectare as well as increased profits by $100-200 per hectare per growing season [19]. Like RiceAdvice, GateMate will be mobile friendly, providing information to farmers in an easily accessible format. While RiceAdvice provides guidance similar to what GateMate intends to deliver, actions will automatically be taken based upon the guidance. In the case of GateMate, the gates will be moved up or down to modify water levels based on guidance from information on weather, topology, and best growing conditions rather than simply providing advice to the farmer to himself implement the prescribed course of action. This automation will allow farmers to easily achieve cost savings while reducing labor.

Another system developed out of a University in Thailand also uses a mobile application to help farmers with their rice farming. This application focused on providing basic information for farmers as well as information from a pH sensor and temperature sensor that would automatically monitor rice water quality [20]. This system has quite different functionality than the one that will be built, but both have the same underlying purpose of aiding rice farmers. The project developed by the University in Thailand communicated with the mobile application and a cloud database [20]. Other teams have already developed much of the GateMate's hardware, so this aspect of the project will be primarily focused on the mobile application portion and utilize the pre-built hardware to operate in a similar way to the device from the University of Thailand. The GateMate software will also contain a much broader scope, incorporating the recommendations and automations into a more advanced software that accounts for weather, topology, and best growth practices to most effectively make quality recommendations and automate the hardware accordingly based on the recommendations from the data.

A mobile app developed by ICAR-NRRI, or the Indian Council of Agricultural Research - National Rice Research Institute, called *riceXpert* allows farmers to see information in real time on "insect pests, nutrients, weeds, nematodes and disease-related problems" and more [21]. Farmers can use the application as a diagnostic tool in the field and find quick solutions to issues that arise by "sending text, photo and recorded voice" [21]. This app focuses more heavily on diagnosing issues after they arise instead of helping the farmers to take preventative measures. While *riceXpert* can be a powerful tool for farmers, providing an application such as GateMate can help prevent these issues from occurring, specifically when it comes to incorrect water levels affecting the production from the field. Both the *riceXpert* and the GateMate applications will provide farmers real-time information, but GateMate will allow the farmer to manually change the gate height to distribute water throughout the field or change the height automatically rather than just provide information to the farmer about solutions to improve their farming technique.

While there are various similar applications to aid rice farmers in their labor-intensive work, nothing quite compares to the GateMate device. The software for these gates allows for automated water level control based on data from weather information, best growth conditions for the plants, and the topology of the field. This automation will not only save farmers time in going out to manually control the gates but also save them money by eliminating excess water usage and ensuring prime conditions for growth to produce the maximum quantity from each field. The combination of multiple factors in this software will allow GateMate to be an effective aid to farmers more so than any of the aforementioned applications.

4.0 Approach and Design

4.1 Requirements, Use Cases, and Design Goals

Requirements

- Provide a detailed view of farm field map with topographical data
- Determine and identify the precise placements of the GateMates to optimize water management
- Visually display all GateMate placements for future modifications

- Access weather information for the appropriate respective location so that the GateMates will respond to real time rainfall in order to more evenly and effectively distribute rainwater throughout the field
- Account for conditions of plant growth to distribute the water more effectively
- Wirelessly control GateMate devices from remote locations with any smart device
- Allow GateMate to automatically make real-time modifications to the height of the gate based on conditions, or allow the user to manually modify the height
- Allow users to view water levels at each gate
- Be able to control GateMate devices by user-defined groupings

Use Cases

- Wireless control will allow farmers to control gates without having to go out to the field to change the gate heights
- Determining the most effective placements for the gates in terms of water management will allow farmers to place gates only once without having to move them regularly to find the best placement
- Utilizing weather information, crop growth conditions, and current water levels to determine optimal heights for the gates will remove guesswork for farmers in terms of what height to set the gates at
- Allowing farmers to see water levels will allow them to use real-time data to make decisions on raising or lowering gates manually if necessary

Design Goals

- Modularize code on crop growth information and on positioning of gates so that specific modules can be further optimized by agricultural experts
- System should be resilient enough that if a gate is taken offline for maintenance the network of other gates remains perfectly functional
- Mobile application functionality should be powerful yet intuitive

4.2 High Level Architecture

The GateMate includes various components that must all work together in unison to create an effective solution. The electrical hardware component has been completed, but the mechanical engineering design and software design have yet to be developed. Specifically, the GateMate software must communicate with a user and the hardware that is controlling the device in the field. Due to the complex nature of the problem, the architecture of the software must be well designed to ensure the successful implementation of all required tasks.

The GateMate software will be developed using React Native, allowing the software to be suitable for any device a user might own, including iOS, Android, or Windows devices. This software will interact with hardware built previously, including a MSP430 microcontroller and a ESP8266 Wi-Fi module, which will provide the interface for connecting the GateMate devices to the GateMate software. A high-level design implementation architecture highlighting the application design process is shown below in Figure 1. The overarching business logic involved with processing gate and geographical data in order to determine gate locations will be discussed first.

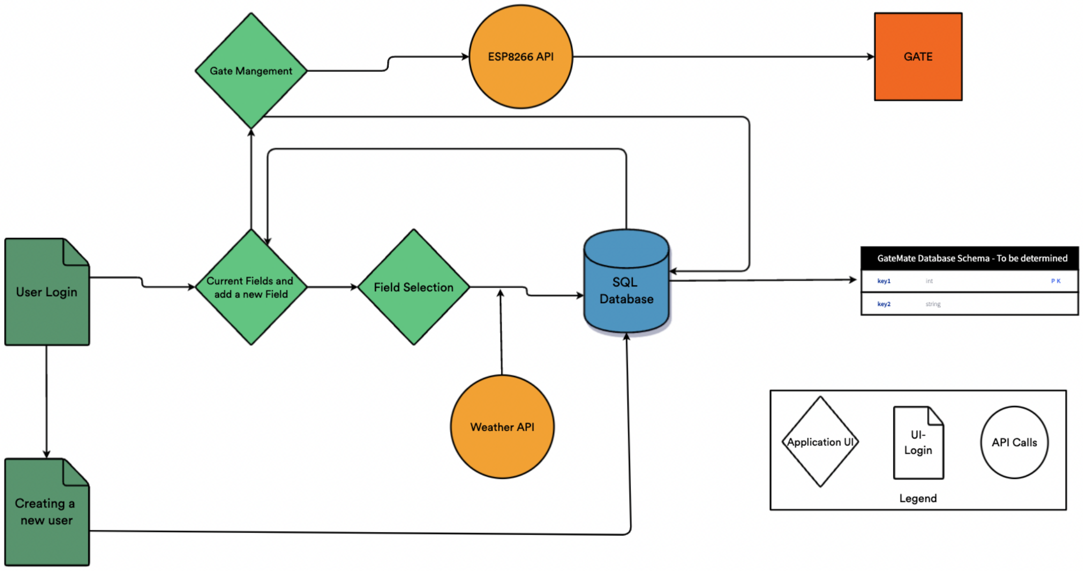

Figure 1: GateMate Application Architecture

Once the gate is connected to the GateMate software, data from the water sensor as well as current gate height information can be used to provide suggestions on appropriate gate height. Additionally, this application will interact with a server storing database information for the weather data, the water level sensor data, data about the current states of the various GateMate devices in the fields, and user information along with data about their fields and gate locations. A weather API will be utilized to collect weather information for the respective field locations, being checked periodically to stay up to date with changes in the forecast. A robust weather and climate API is already readily accessible through Google Earth Engine, a geospatial processing service, which contains a petabyte-scale catalog of free-to-use geospatial datasets. This includes access to weather datasets that include forecast information dealing with precipitation, temperature, humidity, and wind. Climate models are also present, which can help generate long-term plans of action. The database schema will involve analyzing the data available through the Google Earth Engine API as well as the necessary weather data for leveraging rainfall to efficiently conserve and distribute water throughout a field.

The location of these gates will be calculated and saved in a SQL database along with the necessary user login credentials and data for field locations and weather. Each individual gate and its corresponding height will also be saved in the database. Server-side algorithms will leverage this data to determine the optimal placements for gates in a given field location based on topography, when to adjust the height of the gate based on conditions for best growth, and when to adjust the height of the gates based on weather changes such as forecasted rainfall. Each of these will be broken into modules to enable further optimizations, such as machine learning algorithms, that can be implemented at a later time once more data is collected concerning the optimal water levels for given conditions.

Establishing a connection between our application and the ESP8266 Wi-Fi module attached to the present GateMate prototype will be necessary to remotely control the gate from a smartphone. This implementation process will involve the use of the ESP8266 API, which provides the basic software and hardware resources that simplify application development [20].

As for the user interface (UI), it will be simplistic and easy to use while providing the broad range of functionality illustrated in the following figures. Ideally, the map present in the UI would be able to generate a satellite view. With a navigation bar, the user will have various options to change to another view relating to information about the selected field. The user can change views, represented below by a water droplet icon, to see current gate status, their locations, and their current heights. Another tab indicated by the cloud will show suggested modifications to gate heights based on forecasted weather. A third tab represented by a satellite will take the user to a screen that allows them to add a field and generate suggested locations for gates. The final tab will be a settings tab, which will allow a user to configure ideal water levels for best growth conditions based on the specific crop variety so that the height of the gates will be modified accordingly.

Upon creating an account or logging into an existing user account with no associated fields, a user will see a screen with a map, allowing them to select and draw the location of their field on the map (Figure 2). The location and boundaries of a field will be determined in a similar manner utilized by Delta Plastics *Pipe Planner* application, in which boundaries of a particular field are defined using the “Draw option” as shown in Figure 3 [16]. A potential add-on feature can include the ability for the user to upload a shapefile, which is a geospatial vector data format for geographic information software, instead of manually drawing the fields themselves. After making a field selection, the number of gates required, and their optimal placements will be calculated in the backend. The gate locations will be populated to the screen as an icon and with specific location information for the user to see and modify as they see fit.

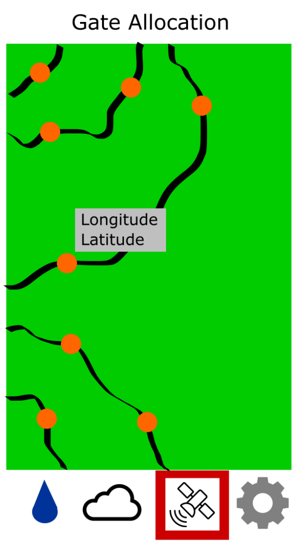

Figure 2: Field Selection View

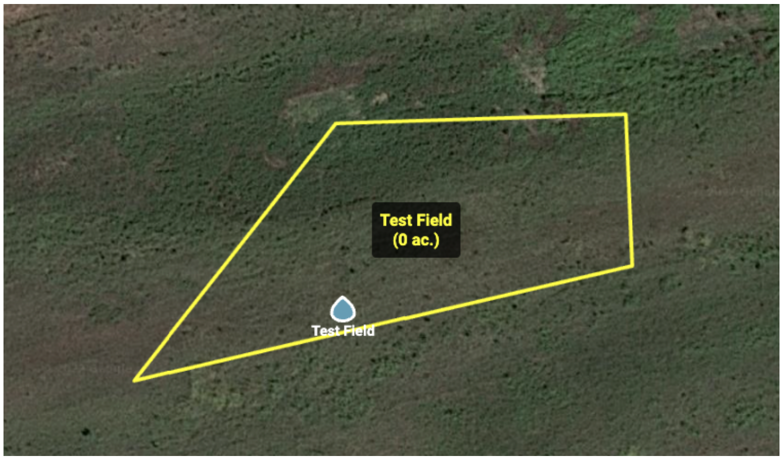

Figure 3: Pipe Planner Field Creation [16]

Once the user has selected their field and confirmed that the gate locations are acceptable, this information will be saved to their account. Upon logging in again, the user will see their saved fields and be able to select which field to look at. Upon clicking on a field name, the user will be able to see the field and gate locations on a topographical map, along with options to raise or lower the height of the gate as shown in Figure 4.

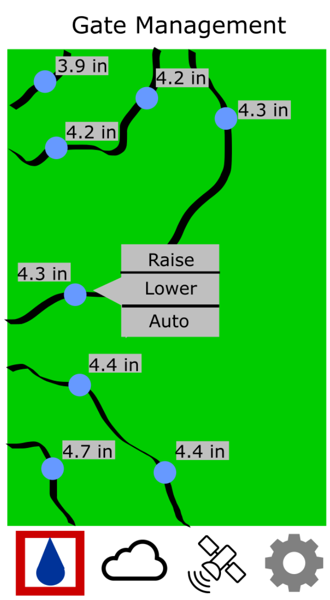

Figure 4: Gate Adjustments

The high-level architecture and design choices are subject to change based on user needs. The proposed application architecture takes into account the different UI views that will need to be implemented as well as the necessary API and database calls in order to fulfill all given requirements.

4.3 Risks

Risk	Risk Reduction
Circuit on gate depletes battery too fast	Optimize code to allow the processor to sleep or stay in a low-power mode as often as possible
Lose connection to a gate	Notify users of a lost connection. Provide information to describe where the gate connection was lost
Farmer places gates in location other than suggested	Allow user to manually select where the gate was placed as opposed to the suggested location
Map information not updated for recent topography changes	Allow the user to confirm the map matches the field topography
Sensors stop working (break, become blocked, disconnected)	Have the software on the gates detect when a sensor might be broken. Have the gates then send a notification indicating a possible problem

4.4 **Tasks**

1. Learn about MSP430 Development and what methods exist for connecting an MSP430 to a mobile device for controlling the gates
2. Confirm with Mr. White that the application design will complete all tasks required
3. Confirm software is compatible with and accounts for all requirements from the Mechanical Engineering device design
4. Establish a database schema
5. Complete final proposal document
6. Complete final proposal slides
7. Create the central server where all data will be stored and connect APIs to it
8. Create a basic application in React Native that connects to the PCBs from the Electrical Engineering team's design
9. Create logic that will get current water levels from the sensors on the device
10. Create basic user interfaces for gate management, home, and settings
11. Create module for logic to determine gate placement
12. Create module for logic to determine gate height
13. Test module to determine gate placement
14. Test module for logic determining gate height

15. Connect server to user interfaces
16. Create functionality for users to draw and select a field area on the map
17. Create functionality for users to select multiple gates
18. Create functionality for users to manually raise/lower gates
19. Create functionality for users to manually place gates
20. Connect logic modules to the respective user interfaces
21. Add error checking to notify user of broken sensors, disconnect from network, or other errors
22. Test connection between software and hardware
23. Test edge cases on software to ensure it operates as expected
24. Document final results and expected operation

4.5 Schedule

Tasks	Tentative Dates
1. Learn about MSP430 Development and what methods exist for connecting an MSP430 to a mobile device for controlling the gates	10/24-11/7
2. Confirm with Mr. White that the application design will complete all tasks required	10/24-11/25
3. Confirm software is compatible with and accounts for all requirements from the Mechanical Engineering device design	10/24-11/25
4. Establish a database schema	10/31-11/4
5. Complete final proposal document	10/31-11/25
6. Complete final proposal slides	10/31-11/25
7. Create the central server where all data will be stored and connect APIs to it	1/15-4/1
8. Create a basic application in React Native that connects to the PCBs from the Electrical Engineering team's design	1/15-4/1
9. Create logic that will get current water levels from the sensors on the device	1/15-4/1
10. Create basic user interfaces for gate management, home, and settings	1/15-4/1

11. Create module for logic to determine gate placement	1/15-4/1
12. Create module for logic to determine gate height	1/15-4/1
13. Test module to determine gate placement	1/15-4/1
14. Test module for logic determining gate height	1/15-4/1
15. Connect server to user interfaces	1/15-4/1
16. Create functionality for users to draw and select a field area on the map	1/15-4/1
17. Create functionality for users to select multiple gates	1/15-4/1
18. Create functionality for users to manually raise/lower gates	1/15-4/1
19. Create functionality for users to manually place gates	1/15-4/1
20. Connect logic modules to the respective user interfaces	1/15-4/1
21. Add error checking to notify user of broken sensors, disconnect from network, or other errors	1/15-4/1
22. Test connection between software and hardware	1/15-4/1
23. Test edge cases on software to ensure it operates as expected	1/15-4/1
24. Document final results and expected operation	4/1-4/28

4.6 **Deliverables**

- Design document: contains a list of the major software and hardware components utilized along with how the software interacts with the mechanical engineering team's physical GateMate prototype that will be built concurrently with the software.
- React Native code: the software used to tell the gates to raise and lower, including the code for the various user interfaces and logic modules.
- Database schema and initial data: data used for the logic modules. Tasks for this semester will include determining an appropriate schema and what sort of database to use.
- Gate microcontroller code: code that runs on the circuit that controls the gate. This software is responsible for all functionalities present at an individual gate. This includes communication using the ESP8266 Wi-Fi module, handling the raising and lowering of the gate via the attached motor, reading data from the attached sensors, and sending collected sensor data.
- Final report: report detailing the completed work on the GateMate software and how it fulfilled the required objectives.

5.0 Key Personnel

Students

Jackson Bullard - Bullard is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed courses in big data analytics, computer networks, mobile computing, database management systems, and cryptography. He has conducted research using distributed machine learning techniques and advanced privacy-preserving technologies.

Nathaniel Fredricks - Fredricks is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has worked two summers as a web development intern at J.B. Hunt and one summer as a custom development intern at Affirma that. He has conducted research related to embedded systems.

Jose Martinez - Martinez is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed coursework in software development and database management. Martinez has relevant work experience at the University of Arkansas as a Research Lab Assistant for the AI and Computer Vision Lab and as a Teaching Assistant for the Programming Paradigms Course. Martinez also completed a Software Engineer Internship with Oracle in Summer 2022, successfully gaining experience in backend microservices and database work.

Carissa Patton - Patton is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed courses including Software Engineering and Wearable and Ubiquitous Computing which will provide a solid background for the software development part of the application. She has also included courses such as Low Power Digital Systems and Circuits and Electronics which allows for understanding of hardware components of the overall systems. Additionally, Patton has completed a Software Development Internship at ArcBest in Summer 2021 and an IT internship at Phillips 66 in Summer 2022.

Ivris Raymond - Raymond is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant coursework in both Software Engineering, Embedded System Security, and Low Power Digital Systems. She has also conducted research related to embedded and IoT systems and completed an internship at Phillips 66 in summer 2022.

Project Sponsors

Jason Bailey - Bailey is the Senior Design Project Coordinator for the Mechanical Engineering Department at the University of Arkansas. He has been the consultant for the mechanical engineering of the gate, especially with regards to the non-computational hardware responsible for controlling the gate, including sensors, servos, and more.

Matthew Patitz - Patitiz is an Associate Professor in the Computer Science & Computer Engineering Department at the University of Arkansas. He will help with scope declaration, help in resolving any issues that might arise with communication with other sponsors, and will provide guidance on any challenges that arise in the implementation of software if necessary.

Timothy White - White is an agriculturist and conservationist who proposed the idea of this GateMate project. From his experience with extensive labor in an agricultural context relating to regulating water for fields, he proposed the GateMate as a way to decrease the labor required and conserve water by removing the guesswork on when to regulate water levels. White has been the expert relating to anything regarding farm or plant specific questions, as well as providing guidance as to what features the final software application should include.

6.0 Facilities and Equipment

The Mechanical Engineering team will concurrently create a GateMate device that will interact with the proposed created software. They work in the mechanical engineering building on campus, and close collaboration between the two teams will be necessary to ensure holistic success. An electrical engineering team from a previous semester has created hardware that will raise and lower the gates of the mechanical engineering design, and our proposed project will interact with this hardware. Specifically, this hardware is a MSP430 microcontroller and an ESP8266 Wi-Fi module, both of which are further discussed in section 3.1.

7.0 References

[1] The Importance of Rice,
http://www.knowledgebank.irri.org/ericeproduction/Importance_of_Rice.htm

[2] Arkansas Rice Facts,
https://www.arfb.com/pages/arkansas-agriculture/commodity-corner/rice/

[3] Total U.S. Rice Production Value from 2000 to 2021,
https://www.statista.com/statistics/190474/total-us-rice-production-value-from-2000/

[4] Hardke, Jarrod. "Arkansas Rice Production Handbook." Little Rock, Ark: Cooperative Extension Service, University of Arkansas, 2013.

[5] Sustainable Agriculture and Water,
https://www.nature.org/en-us/about-us/who-we-are/how-we-work/finance-investing/naturevest/sustainable-agriculture-water/

[6] Water Use by Rice,
https://rice.ucanr.edu/Water_Use_by_Rice/

[7] Using Alternate Wetting & Drying (AWD) Rice Flood Management,
https://www.uaex.uada.edu/publications/pdf/FSA62.pdf

[8] Google, Earth Engine Data Catalog,
https://developers.google.com/earth-engine/datasets

[9] "What is a mesh network?" Google Nest Help.
https://support.google.com/googlenest/answer/7182746?hl=en

[10] "What is an API?" University of California San Francisco, n.d.
https://developer.ucsf.edu/what-api.

[11] "Wi-Fi — ESP8266 RTOS SDK Programming Guide." espressif. n.d.
https://docs.espressif.com/projects/esp8266-rtos-sdk/en/latest/api-reference/wifi/esp_wifi.html.

[12] "React Native Learn Once, Write Anywhere." Reactnative. n.d.
https://reactnative.dev.

[13] "What is a database?" University System of Georgia. n.d.
https://www.usg.edu/galileo/skills/unit04/primer04_01.phtml#:~:text=A%20database%20is%20a%20collection%20of%20information%20organized%20to%20provide%20efficient%20retrieval.

[14] "What is SQL?" IBM. March 3, 2021. https://www.ibm.com/docs/en/developer-for-zos/14.0.0?topic=support-what-is-sql.

[15] "The Power of Automation in Rice Production." Columbia Okura. Columbia Okura, n.d. https://columbiaokura.com/thought-leadership/power-automation-rice-production/.

[16] Revolution. "Pipe Planner." Delta Plastics Home, n.d. https://www.deltaplastics.com/pipe-planner.

[17] Boyd, Vicky, "Siri, Turn Off Pump: Research Examines Potential Water Savings of Automated and Remote Irrigation Systems," Rice Farming, March 10, 2021. https://www.ricefarming.com/departments/feature/siri-turn-off-pump/.

[18] Choudhary, Vikas, and Karin Fock, "Precision Agriculture for Smallholder Farmers in Vietnam: How the Internet of Things Helps Smallholder Paddy Farmers Use Water More Efficiently." World Bank Blogs, April 23, 2020. https://blogs.worldbank.org/eastasiapacific/precision-agriculture-smallholder-farmers-vietnam-how-internet-things-helps.

[19] "RiceAdvice." AfricaRice. The Consortium of International Agricultural Research Centers, n.d. https://www.africarice.org/riceadvice.

[20] Kularbphettong, Kunyanuth, Wannaporn Phoso, and Pattarapan Roonrakwi. "The Automation of Mobile Application to Manage the Rice Fields." TEM Journal: Technology, Education, Management, Informatics 8, no. 3 (August 2019): 866–71. https://www.temjournal.com/content/83/TEMJournalAugust2019_866_871.pdf

[21] "ICAR-NRRI Developed Mobile App 'RiceXpert.'" National Rice Research Institute. ICAR-National Rice Research Institute , n.d. https://icar-nrri.in/icar-nrri-developed-mobile-app-ricexpert/.