Report on the SCRUM and SPRINT

CEG 4912

Submitted by: Group 2

Aya Chatiou, 300203768
Mayssa Tebourbi, 300145817
Esdras Sumaili, 300210658
Nujhat Fatima, 300130310
Sultan Oloyede, 300076997
Decaho Gbegbe, 300094197

Author Department: Faculty of Engineering
University of Ottawa

Sprint 1

Week 1 (Jan 13th - Jan 20th)	**Project Planning & Proposal** ● Software architecture model rough draft ● Hardware architecture model rough draft ● Requirements review ● Component query ● WBS rough draft ● Github setup ● Trello board setup ● Gantt chart set up
Week 2 (Jan 21st - Jan 27th)	**Hardware Setup** ● Component acquisition ● Raspberry Pi OS setup ● Python Libraries (GPIO pin, rplcd.i2c, pi camera etc.) prepared for system ● Initial set up of wires ● Virtual environment setup
Week 3 (Jan 28th - Feb 3rd)	**Sensor and Screen Implementation** ● Initial sensor integration with MCU to prepare for object detection ● Initial screen integration with MCU to print states of sensor. ○ “System ready” - system is ready for mail insertion ○ “Object detected” - sensor detects object ○ ”Object captured” - Camera takes picture
Week 4 (Feb 4th - Feb 10th)	**Camera Implementation** ● Initial camera integration with MCU to prepare for objects to be captured once the sensor detects an object ○ Issues faced: pip command not working, unable to download tesseract library ○ Possible solution: manually download libraries **Midterm Presentation (Feb 9th) Preparation** ● Project presentation slides prepared ● Final draft of WBS created ● Final draft of Software architecture model ● Final draft of Hardware architecture model
Deliverable 1 Demo - Feb 13th	**Demo of** ● Sensor implementation ● Screen implementation ● Camera implementation **Summary of demo:** System was initialized by creating a python script of the LCD screen with its system ready state and once an object was detected, the camera took a picture of the object.

Sprint 2

Week 1 **(Feb 26th - March 3rd)**	**Mid-Session Report Completion:** ● Finalized and submitted the mid-session report on February 26th. ● Conducted thorough review and edits before submission. **SPRINT and SCRUM Explanatory Report:** ● Completed the write-up on SPRINT and SCRUM methodologies by February 28th. ● Illustrated the application of these methodologies to our project. **Bug Fixing and Troubleshooting:** ● Identified and documented bugs from Sprint 1, focusing on sensor object detection issues. ● Debugged the sensor-to-Raspberry Pi integration to correct the erroneous 'no object detected' message. ● Tested and validated the sensor's accuracy post-troubleshooting. **Hardware Enhancement Planning:** ● Researched and selected a suitable breadboard for component integration. ● Planned the procurement of the breadboard to facilitate a more streamlined hardware setup.
Week 2 **(March 4th - March 10th)**	**AI Calibration of Letter Address Reading:** ● Download tesseract & opencv library ● Research OCR Process Flow ● Implement OCR algorithm to detect ○ orientation of text ○ digits ● Collect data for different handwriting **Component Integration onto Breadboard:** ● Insert the sensor and screen modules into the breadboard, ensuring secure and correct connections. **Integration Testing with Existing Code:** ● Run the pre-existing sensor and screen interaction code to validate the new breadboard setup. ● Troubleshoot any connectivity issues and confirm the seamless operation of integrated components.
Week 3 **(March 11th - March 17th)**	**AI Calibration Continues:** ● Implement OCR algorithm to detect ○ orientation of text ○ digits ● Collect data for different handwriting ● OCR Testing ○ System integration testing ○ Unit tests ○ Model testing **Motor Configuration and Integration:** ● Prepare wiring diagrams for connecting the motor to the Raspberry Pi. ● Physically set up the motor and ensure correct electrical connections. ● Integrate the motor with the Raspberry Pi's GPIO pins for control signals.

	Python Script Development for Motor Testing: ● Utilize the RPi.GPIO library for interfacing with the motor. ● Write a Python script to control motor actions, aiming to selectively pull out letters. **Independent Component Operation Testing:** ● Test the motor's functionality independently from other system components. ● Document the test results and adjust the motor control parameters as necessary.
Week 4 **(March 18th - March 24th)**	**AI Calibration continues:** ● OCR implementation continues ● OCR Model testing **Machine Learning Model Training:** ● Enhance the letter reading and detection capabilities through additional machine learning training sessions. ● Refine the dataset and adjust model parameters to improve accuracy in letter detection. **Motor and Component Integration:** ● Integrate the motor within the overall system to function with other components regardless of the item type. ● Test the motor's interaction with the system, ensuring it operates under the command of the MCU. **Letter Holder Setup:** ● Design or procure a suitable holder or compartment for the placement of letters. **LCD Screen Configuration for Error Handling:** ● Program the LCD to display a 'Wrong Insertion' message when an incorrect object is detected. ● Sync the display of this message with the activation of the motor to eject the wrongly inserted object.
Week 5 **(March 25th - March 31st)**	**AI Calibration Continues:** ● Camera should detect typed text ● Continue training model for handwritten texts **Back-end Operations begin:** ● Database Initialized ● ER schema and diagram **Front-end Operations begin:** ● Rough draft of GUI ● Use cases ● FSM ● High level view of front-end prepared **Camera Functionality and Validation:** ● Check the system's logic to accurately discern if the captured image is a letter, prompting 'Letter Captured' on the LCD. **Error Detection and Motor Response:** ● Ensure the system identifies non-letter items, displaying 'Wrong Insertion' on the LCD. ● Test the motor's ejection mechanism to reliably reject items when a non-letter is detected.
Week 6 **(April 1st - April 7th)**	**Notification Alert System initialized:** ● Back-end & Front-end integration **Second deliverable system integration testing:**

	● Testing OCR model to detect clean handwriting text ● OCR model should detect written text ● All components implemented functions as expected **Comprehensive System Testing:** ● Conduct full-scale tests to ensure all components (sensor, camera, LCD, motor) are functioning as expected. ● Validate the first stage of the sorting process, confirming flawless operation. **Prepare Final Presentation** ● Prepare Report ● Prepare presentation slides ● Prepare Demo

Sprint 3

Week 1 (Sep 4 - Sep 8)	**Strict text recognition for apartment numbers** ● Improve the Optical Character Recognition software OCR to read and interpret the apartment numbers or letters.
Week 2 (Sep 9 - Sep 15)	**Display apartment numbers on LED** ● Integrate the OCR output with the display system. ● Present visually the detected apartment number, for verification or control purposes.
Week 3 (Sep 16 - Sep 22)	**Synchronize the motor to the apartment box** ● Involve programming and mechanical work to ensure that once the apartment number is recognized, the system's motor will correctly sort the letter and push it to the corresponding box.
Week 4 (Sep 23 - Sep 29)	**Build a motorized platform to sort letters** ● Design the hardware platform that physically moves the letters ● Integrate it with the motors, that will be controlled by the system's software
Week 5 (Sep 30 - Oct 6)	**Initialize the database for recipient details** ● Create a database to store information about the recipients, such as names, apartment numbers, and any other relevant details that could be used for sorting or record-keeping ● Create a messaging system and make it able to notify the clients when they receive a letter in their box.
Week 6 (Oct 7 - Oct 13)	**Finalize the second floor of the mailsorted box** ● Link the database with the messaging system ● Be able to demo the operation of the second floor

Sprint 4

Week 1 (Oct 21th - Oct 27th)	**Project Testing and documentation** ● Unit testing : test if the assembly unit works correctly with all libraries correctly used ● Integration testing : see if the unit is correctly installed and solder in the device in order to be completely stable ● check that the LCD, the motor, the sensor, the hatches and the support are all correctly assembled and soldered that the device, although portable, supports all the elements
Week 2 (Oct 28th - Nov 3rd)	**Refine system** ● Assembling with the AI so that it effectively carries out these different tasks ● Correct the system by check that the LCD correctly displays all status messages based on sensor detection in order to verify that they are perfectly synchronized ● based on testing results: Address any bugs or functionality issues identified during testing. ● Finalize documentation: Ensure user manuals and troubleshooting guides are complete and polished.
Week 3 (Nov 4th - Nov 10th)	**Finalize hardware and software:** ● Finalize any hardware/software modifications based on testing and user feedback. ● Check system performance to ensure that the device rejects if the detected object is not a letter and efficiently sorts the detected letter into the correct compartment
Week 4 (Nov 11th - Nov 17th)	● **Running the mailbox and finalizing the assembly** to ensure it is ready for delivery to the customer
Week 5 (Nov 18th - Nov 24th)	**Prepare final presentation** ● Develop a comprehensive presentation showcasing the project's goals, achievements, and future potential. ● Prepare for the final presentation and demonstration.
Week 6 (Nov 25th - Dec 1st)	**Practice presentation** ● Rehearse the final presentation to ensure clarity and effectiveness.
Week 7 (Dec 2nd - Dec 8th)	**Final presentation and demonstration**: ● Deliver the final presentation and showcase the SSM-ARMS system's functionality.
Week 8 (Dec 9 - Dec 15)	**Project wrap-up**: ● Finalize all project documentation and reports.