

# ARRM: An Objective Faculty and Research Assistant Matching System

Mercy Onyinyechukwu Chimezie  
Computer Science,  
Ashesi University  
Accra, Ghana  
mercy.chimezie@ashesi.edu.gh

Ayodeji Oluwapelumi Falase  
Computer Science,  
Ashesi University  
Accra, Ghana  
ayodeji.falase@ashesi.edu.gh

Richard Nana Kweenu Quayson  
Computer Science,  
Ashesi University  
Accra, Ghana  
richard.quayson@ashesi.edu.gh

Ruvimbo Joy Sithole  
Computer Science,  
Ashesi University  
Accra, Ghana  
ruvimbo.sithole@ashesi.edu.gh

**Abstract**— Research is central to every organisation as it helps identify trends and inform policy. However, the process of selecting or matching research assistants can be quite daunting for various reasons. The software this paper develops will act as an information management system on contracted RAs, their availability, personal interests, expertise, and relevant past work and experiences. The software will allow faculty members to create bios, submit research proposals and request RAs for such proposals. The software will equally provide functionalities that will enable the Research Studio and faculty alike to create sub-tasks and monitor and review the progress of the assigned RA. Finally, the software will feature an anonymised rating system to allow RAs and faculty to rate each other, to be utilised in the pairing process.

**Keywords**—smart matching, task management, information system, progress tracking, blind feedback.

## I. INTRODUCTION

Research is central to every organisation as it helps identify trends and inform policy. This influence is even so in educational institutions and academia in general, as such educational institutions are the primary contributors of knowledge to existing research. However, the process of selecting or matching research assistants can be quite daunting for various reasons. Ashesi University, an educational institution, faces similar challenges where it is currently challenging to match research assistants with faculties on their various research topics. As such, this project aims to smooth line the matching process by creating a simple and efficient platform where faculty can request help on a research project and research assistants can easily collaborate on faculty research projects. The application will support straightforward registration, smart project searching, tracking research work progress, simplified research findings sharing, and feedback options for both research assistants and faculty.

## II. LITERATURE REVIEW

A critical aspect of a Research Assistant Matching system is task management, as tasks will have to be created, managed, prioritised, tracked and completed between Research Assistants (RAs) and Faculty members. There is an increasing need for efficiency and effectiveness in performing and completing tasks on time as the ability to organise, delegate and schedule tasks on this application will determine the work success between RAs and faculty members. This paper [5] focuses more on task prioritisation as a significant way to achieve work goals. A task matrix, specifically the Eisenhower matrix, is emphasised. "It is based on the Eisenhower Method of arranging tasks by urgency and

importance in a 2×2 matrix in which the key features or functions of the project deliverable are listed and prioritised according to their importance." [5]. In the research assistant matching system, this matrix, when integrated, can enable research assistants to prioritise tasks and enable faculty to have an understanding of how engaged the research assistants are. The second paper [2] on task management looked at existing challenges with the current task management systems. It provided some necessary design requirements that can be incorporated into task management systems to make them more effective. Some requirements critical to the research assistant matching system include tracking, scheduling, time management recorder, notifications and reminders, task completion and feedback [2].

From the review of the two pieces of literature above, a task management system is essential to the research assistant matching system as it would help research assistants and faculty have a better experience in their work. The challenge with the first paper is that it only focused on one aspect of task management, which is task prioritisation, and this does not give a holistic view of the functionalities of a task management system. The second paper, on the other hand, poses the challenges of other systems, such as " (i) Integrating different types of media used to manage tasks (e.g. email, to-do lists or calendars) in a more insightful and controlled way. (ii) Rearranging tasks due to unexpected situations. (iii) Determining which tasks should or could be done within a certain time-frame or reasonable time-frames. (iv) Scheduling irregular tasks and generating flexible scheduling. (v). Monitoring their current management habits and their actual progress or achievements." [2] Some of these challenges will be addressed by the research assistant matching system.

In the paper, Ren et al. research matching algorithms and discuss two kinds of matching algorithms: explicit matching and implicit matching. "Explicit matching refers to problems in which the agents themselves give preference lists [3]. Thus, the ultimate goal will be to have a one-on-one matching for each agent to pair up with the next available agent on their preference list. This methodology, however, limits the efficiency of the matching process. The research paper uses one-on-one matching of men and women in a marriage market to pair up people into couples as an example. The algorithm considers each agent's preference list of people they are interested in such that an individual can only be paired with their preferred person if they, too, are listed as a preference. The shortfall of this algorithm concerning this paper is that it reinforces bias as faculty members have a tendency to select RAs that they are familiar with despite the RAs' availability or their knowledge in the area. Implicit matching works in

conditions "where the preference list is not provided". The algorithm works by considering each agent's needs/wants and assigning weights to them. The research paper used an example of a search engine finding the best result to return to a user's query; a suitable match for the agent is found by running through the list of candidates to obtain one with the highest total score.

K Liu proposes frontend and backend separation as a way "to reduce the degree of coupling between the model, view and control in the system" [4] so that by guaranteeing orderly processing of research projects, the system could achieve hierarchical management and monitoring of information on each research project process. The MIS is designed to bring together project management, document management, task management, and communication management, which are crucial to research project management. The architectural design stems from the architectural pattern of MVC; frontend and backend separation allow the view and model to be independent so that the frontend comprises the view and controller, and the backend is responsible for the data logic and business processing only. Because of the complexity of the research system in terms of querying information and tracking information gathered, having a separate backend component that handles the logic will allow the computing power to match the user's demands of operating speed. As a result, "several backend cases can be executed at the same time to meet the operation of complex large-scale system." [4] The task of data display and request response is delegated to the frontend component of the system. By separating the frontend and backend, the MIS can be more flexible and scalable, and different frontends can be developed to meet the needs of different users.

### III. RELATED SYSTEMS

#### A. *ResearchGate*

ResearchGate, primarily established as a platform dedicated to the dissemination and retrieval of scholarly publications and fostering robust networking within the research community, shares several noteworthy commonalities with this project. ResearchGate provides a dynamic environment where researchers are not only connected with their peers but can also actively participate in knowledge exchange and collaborative discourse [7]. It offers a comprehensive user experience, enabling individuals to construct intricate profiles showcasing their areas of research interest and actively engaging with like-minded scholars in their field. It empowers researchers to effortlessly explore an extensive repository of publications, identify domain experts, and join specialized research communities, streamlining their access to relevant academic resources [7]. At its core, ResearchGate enables researchers to disseminate their findings via publications and actively partake in discussions, creating a space for scholarly interaction and feedback.

#### B. *PolyRatings*

PolyRatings, a notable website, extends an invaluable platform to students, affording them the anonymity to appraise and assess their instructors [6]. It functions as a medium through which students can furnish comprehensive feedback on their educational encounters, encompassing detailed assessments of their instructors in various facets of their teaching methodologies. The underlying objective of this system is to augment transparency and equip students with the tools to make judicious decisions when choosing their courses

and instructors. At its core, PolyRatings empowers students to establish profiles, which in turn serves as a conduit for the provision of anonymous reviews and ratings for their instructors [6]. This registration process facilitates the seamless collection of valuable feedback from students. One of PolyRatings' key functionalities lies in its capacity to solicit student feedback through instructor ratings and reviews, engendering a comprehensive repository of evaluative data [6]. Within the PolyRatings framework, students wield the power to assess and appraise their instructors, scrutinizing various parameters, encompassing teaching style and effectiveness, mirroring the multifaceted evaluation process. It not only facilitates student reviews but also includes a search feature, enabling students to locate and assess instructors efficiently.

#### C. *Asana*

Asana is a platform for collaboration and a cross-functional platform. Asana is a work management platform that helps teams collaborate and coordinate their tasks and projects. [1]. Asana's functionalities as a project and task management system can be incorporated into the research assistant matching system such that it can be helpful in research work between research assistants and faculty by helping them to create and assign tasks to track the progress of different research activities, such as literature review, data collection, analysis, writing, etc [1]. It can also help with setting goals and linking them to tasks and projects to align the research work with the desired outcomes and measure the impact of the research assistants.

### IV. USER STUDY

The client interviewed Research Assistants (RAs) and Faculty members at Ashesi University on their gains and pain points with the current process of matching RAs and faculty. As such, this paper relies on the secondary data from the interviews conducted by the client and primary data from interviews with the client to understand better the needs of the stakeholders for the project. The following sections describe the insights drawn from the interviews.

The current matching means utilises an Excel sheet containing the details of contracted research assistants, which is created and shared with faculty members via email. From there, faculty members can review the file shared and request for a listed RA from the Research Studio – although this means is seldom used. Other means of matching include faculty reaching out to RAs they had worked with before or had taught back when they were in school directly for their assistance. However, challenges from the knowledge gap on the faculty's part regarding the availability of the RAs, the appropriate means to request an RA and the information required in an RA request stem from the current system. Moreover, due to insufficient tracking of the workload of RAs, faculty are primarily unaware of whether the RAs are preoccupied with other research work or not, leading to workload misunderstandings.

The primary support sought by faculty from RAs includes:

- a) assistance at various stages of research, from topic formulation through manuscript writing, literature review, data collection and analysis to applying to the Institutional Review Board (IRB);

- b) identification of relevant publications and the criteria for research publication; and
- c) requests for sourcing support and research criteria (such as grants, sponsorships, and conferences).

RAs, on the other spectrum, deduce the complexity and workload of the faculty's proposed research, the faculty's reputation, availability, expertise and personal interests when selecting or accepting faculty research proposals.

In light of these insights, the primary goal of this paper is to develop software for the client to aid in the matching of RAs and faculty members. The software will act as an information management system on contracted RAs, their availability, personal interests, expertise, and relevant past work and experiences. The software will allow faculty members to create bios, submit research proposals and request RAs for such proposals. The software will equally provide functionalities that will enable the Research Studio and faculty alike to create sub-tasks and monitor and review the progress of the assigned RA. Finally, the software will feature an anonymised rating system to allow RAs and faculty to rate each other, to be utilised in the pairing process.

One of the critical functionalities of the *ARRM* project is to design an implicit matching algorithm [3] that matches faculty with RAs based on project requirements, expertise, and availability. Stemming from the research, the developers will work with the latter algorithm as it reduces the tendency of faculty to list research assistance based on their familiarity with them, allowing a more accurate pairing based on the researcher's availability, skills, and field of expertise. Additionally, the research assistant and the faculty will be considered valid users of the system whose needs are used in the algorithm to generate preference lists. The system will then match individuals who mutually benefit the most from the connection so that research assistants can work on projects they enjoy more.

The *ARRM* system will adopt a similar strategy [4] to enable quick querying of pending research tasks and available research assistants. The platform will be robust and efficient in providing information that the user requires on the progress of any assigned task. Apart from providing an efficient implementation with low latency, the project will have a strong emphasis on designing a user-friendly interface that is easy to navigate and get accustomed to. The frontend component is crucial to the success of this project as one of the stated pain points from the faculty was their needing to be able to familiarise themselves with the various technical solutions provided for them.

#### A. Sustainable Development Goals

The *ARRM* system supports the following SDGs:

- i. SDG 3: The *ARRM* system supports the third sustainable development goal – "ensure healthy lives and promote well-being for all at all ages" [8] – through its task management features. The task management features aim to help Faculty and RA prioritise, assign, and track milestone tasks to make work more efficient and effective. This promotes healthy living and well-being of RAs and Faculty as they get to work at a moderated healthy pace and maintain all-round

well-being.

- ii. SDG 8: This paper's goal of improving the process of matching Research Assistants and Faculty members aligns with the principles of sustainable development goal eight. SDG 8 seeks to "promote economic growth, employment, and decent work for all, focusing on inclusivity and sustainability" [8]. The projects promote opportunities for research assistants to find meaningful employment by collaborating with faculty on research projects with shared interests.

#### B. Intended Audience

The *ARRM* web-based application has the following stakeholders:

- Faculty members who engage in research projects aside from lecturing.
- Research Assistants who assist faculty members with their research project needs.
- Ashesi University's Research Studio who are responsible for managing research endeavours in the organisation.

### V. SYSTEM REQUIREMENTS

Insights from the literature review, related systems, and user needs garnered various requirements for the *ARRM* system to provide functionality. The PolyRatings [6] and ResearchGate [7] systems informed the project of the need for a blind feedback mechanism and public user profiles. The user needs elicitation process further highlighted the need for these features and provided further needs from an efficient implicit matching algorithm devoid of bias [3] through rotational assignment upon the completion of a project to task management of various tasks under a project by Faculty members and Research Assistants [1], [2].

In implementing the *ARRM* system, this paper utilises the agile methodology. The agile methodology best suits the project due to the changing nature of the requirements. The changing requirements lend the project to adopt a methodology that embraces change and releases versions of the system in increments. Moreover, due to the limited timeframe for the project, covering about a month and a half, extreme programming, test-first development, pair programming, and other such agile methodology principles will enable the project to meet the deadline. The following subsections discuss the user stories influencing the functionalities, the non-functional requirements, and the constraints of the *ARRM* system.

#### A. User Stories

US001: As a Faculty or RA, I want to be able to create comprehensive profiles on my area of expertise, research interests, availability and relevant my area of expertise, research interests, availability and relevant publications and documents so that I can be matched with appropriate Research Assistants in my area of specialisation.

US002: As a Faculty, I want to be matched with other RAs based on my project's requirements, the

RA's expertise, and availability so that they can assist me with various tasks in my research process.

US003: As an RA, I want my profile to be updated after I have been matched with a Faculty's research so that faculty will be informed of my engagement and not conflict with my availability.

US004: As a Faculty member, I want a medium, on-platform, that I can use to initiate contact with recommended RAs so that I can easily keep track of available RAs to work with on a research project.

US005: As a Faculty member, I want to create research projects after providing the requirements, preferred skills of assistants, and estimated hours for the research so that I can be matched with available RAs.

US006: As a Research Assistant, I want to receive notifications when Faculty create new research projects through in-app messaging and email mediums so that I can be aware of ongoing projects and express interest.

US007: As a Research Assistant, I want to be able to search, view and express interest in available research requests posted by Faculty so that I can easily keep track of faculty project needs.

US008: As a Research Assistant, I want to be automatically recommended for ongoing research projects once I complete my current project so that matching with new faculty and research projects will be easier.

US009: As a Faculty, I want to be able to create and assign milestone tasks of my research project, with deadlines, to RAs so that I can keep track of their completion rate.

US0010: As a Faculty or RA, I want to be able to share experiences, suggestions, and questions on ongoing and completed milestone tasks in research projects so that collaboration between us can be made easier and improved.

US0011: As a Faculty or RA, I want to be able to provide overall feedback and review the performance of the assigned Faculty or RA upon the completion of a research project so that it will validate and inform the skills and professionalism of the RA or Faculty.

User Story	Priority
US001: Profile Creation	1
US002: Faculty Matching	2
US003: RA Matching	3
US004: In-app Messaging	4
US005: Project Creation	5
US007: Research Project Requests	6
US009: Task Creation & Assignment	7
US008: Project Recommendation	8
US006: Notification	9
US0011: Review	10
US0010: Feedback Sharing	11

Table 1.0 shows the user stories and the priority (chronological order) in which this project aims to implement the functionalities listed.

The project will implement user stories with priorities 1 to 7 as they are core to the *ARRM* system. However, user stories with priorities 8-11 have medium-level priorities and their implementation enhance the core the functionalities of the *ARRM* system.

## B. Non-Functional Requirements

NFR001: Usability: As a developer, I want the interface of the *ARRM* system to have a very low learning curve by using intuitive and user-friendly interfaces so that the user adopts the platform with minimal training and adoption curve. I want the system to employ ease-of-use heuristics that utilise graphs, charts, and patterns so that the faculty and RA use recognition rather than recall.

NFR002: Safety/Security: As a developer, I want the *ARRM* system to control access to user profile data and project information through authentication and authorisation measures so that the end user's data privacy is maintained, and they only make changes to the information they are accountable for. In addition, I want the system to maintain data privacy and security at all stages of development so that the end user is restricted from making unauthorised alterations to maintain data integrity and consistency.

NFR003: Performance: As a developer, I want the *ARRM* system to respond to user interactions within 5 seconds and support a concurrent user load of at least 50 users without performance degradation so that the user is encouraged to adopt the platform and easily make frequent updates to the platform.

NFR004: Availability and Reliability: As a developer, I want the *ARRM* system to be available 24/7 and have an uptime of at least 99%, so that the user can utilise the functionalities of the platform as of when they need it. Scheduled maintenance will be communicated in advance so that the user plans for downtime.

NFR005: Error Handling: As a developer, I want the *ARRM* system to prioritise Nielson's heuristic on error prevention and display appropriate error messages so that the end user easily recognises, diagnose, and recover from the errors.

NFR006: Data Retention: As a developer, I want the *ARRM* system to retain user profiles, research projects and other application data forever so that the user has their saved preferences, project and profile data loaded upon login. Moreover, the system shall archive data before updating so that the user can easily restore it.

NFR007: Compatibility: As a developer, I want the *ARRM* system to be compatible with major web browsers such as Chrome, Safari and Microsoft Edge so that users can access the system from any browser they have.

NFR008: Scalability: As a developer, I want the *ARRM* system to have a scalable and modular architecture so that the developer can accommodate future enhancements that come with evolving end user needs.

Non-functional Requirement	Percentage (%)
NFR001: Usability	18
NFR002: Availability and Reliability	18
NFR003: Privacy/Security	17
NFR004: Compatibility	12
NFR005: Data Retention	10
NFR006: Performance	10
NFR007: Error Handling	8
NFR008: Scalability	7

Table 2.0 shows the non-functional requirements and the extent to which this project promotes them.

### C. Constraints

#### 1) General Constraints:

- C001: Timeline Constraint: The system must be designed and developed within the specified timeframe to meet the set deadlines.
- C002: Resource Availability Constraint: The limited availability of resources, including human, hardware, and software resources, constrains the development of the *ARRM*.
- C003: User Acceptance Testing Constraint: Regular user acceptance testing to ensure the *ARRM* system meets RA and Faculty expectations is constrained by limited time.

#### 2) Application-specific Constraints:

##### RA and Faculty Account Creation Constraint:

- C004: *ARRM* administrators are responsible for creating accounts for contracted Research Assistants and Faculty members.

##### RA Profile Constraint:

- C005: RAs must upload their CVs and unofficial transcripts to complete their profiles.

##### Faculty-RA Matching Constraints:

- C006: An RA's profile must be complete for the algorithm to recommend them during the matching process.
- C007: Faculty and RA must accept a potential match before the matching process is completed and their profiles are updated with the respective project.

C008: Faculty-RA matching shall consider the maximum number of hours that an RA can work in any given week when pairing RAs with Faculty.

C009: Faculty approval is required for RA participation in research after rotational assignment.

##### Research Project Constraint:

C0010: Faculty approval on work hours and output of RA is required for an RA to complete an assigned research project.

##### Blind Faculty-RA Rating Constraint:

C0011: Faculty and RAs can review RAs and Faculty respectively only after the completion or termination of a research project.

## VI. SYSTEM DESIGN

This chapter presents the *UML* diagrams created for the *ARRM* system. Please refer to the *Appendix* section for the process flow, sequence, state, activity, use case, class, object, and context diagrams.

## REFERENCES

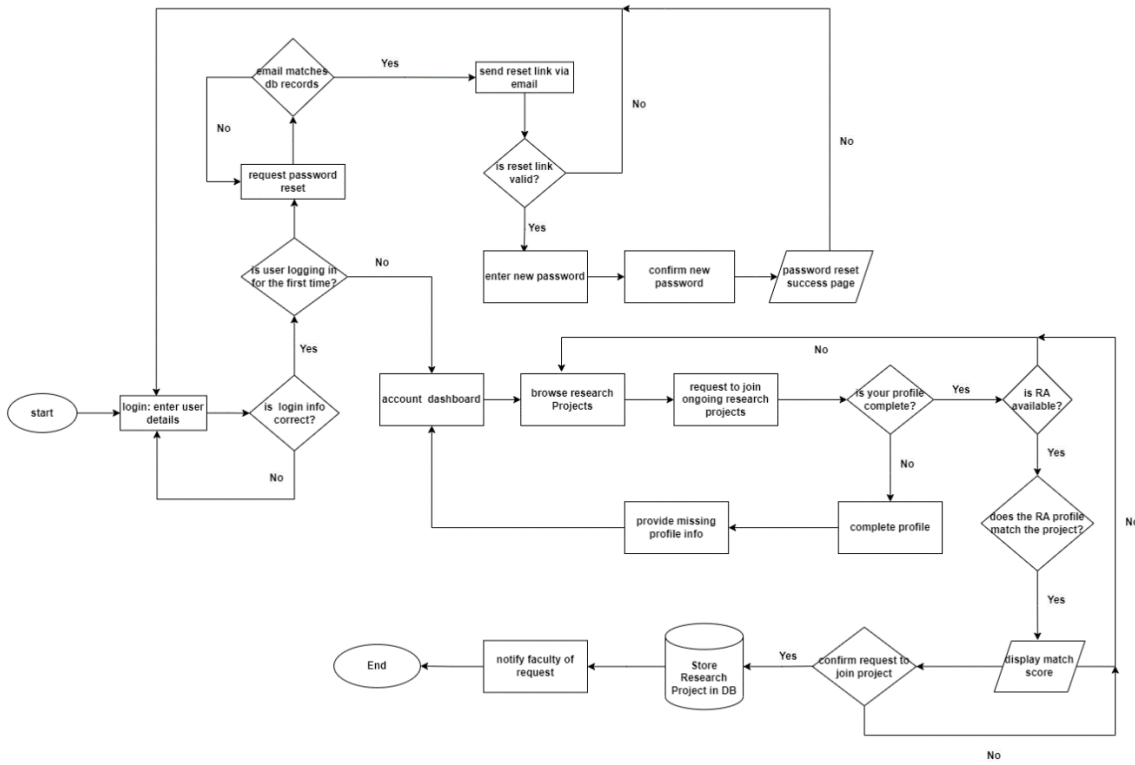
- [1] Asana, "Manage your team's work, projects, & tasks online • Asana," Asana. Accessed: Oct. 31, 2023. [Online]. Available: <https://asana.com/>
- [2] A. Kamsin, A. Blandford, and A. L. Cox, "Personal task management: my tools fall apart when I'm very busy!," in *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, in CHI EA '12. New York, NY, USA: Association for Computing Machinery, May 2012, pp. 1369–1374. doi: 10.1145/2212776.2212457.
- [3] J. Ren et al., "Matching Algorithms: Fundamentals, Applications and Challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 3, pp. 332–350, Jun. 2021, doi: 10.1109/TETCI.2021.3067655.
- [4] K. Liu, J. Jiang, X. Ding, and H. Sun, "Design and Development of Management Information System for Research Project Process Based on Front-End and Back-End Separation," in *2017 International Conference on Computing Intelligence and Information System (CIIS)*, Apr. 2017, pp. 338–342. doi: 10.1109/CIIS.2017.55.
- [5] N. S. Jyothi and A. Parkavi, "A study on task management system," in *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, May 2016, pp. 1–6. doi: 10.1109/RAINS.2016.7764421.
- [6] "Polyratings.com." Accessed: Oct. 31, 2023. [Online]. Available: <https://www.polyratings.com/>
- [7] "ResearchGate | Find and share research," ResearchGate. Accessed: Oct. 31, 2023. [Online]. Available: <https://www.researchgate.net/>
- [8] "THE 17 GOALS | Sustainable Development." Accessed: Nov. 07, 2023. [Online]. Available: <https://sdgs.un.org/goals>

## APPENDIX

### A. UML DIAGRAMS

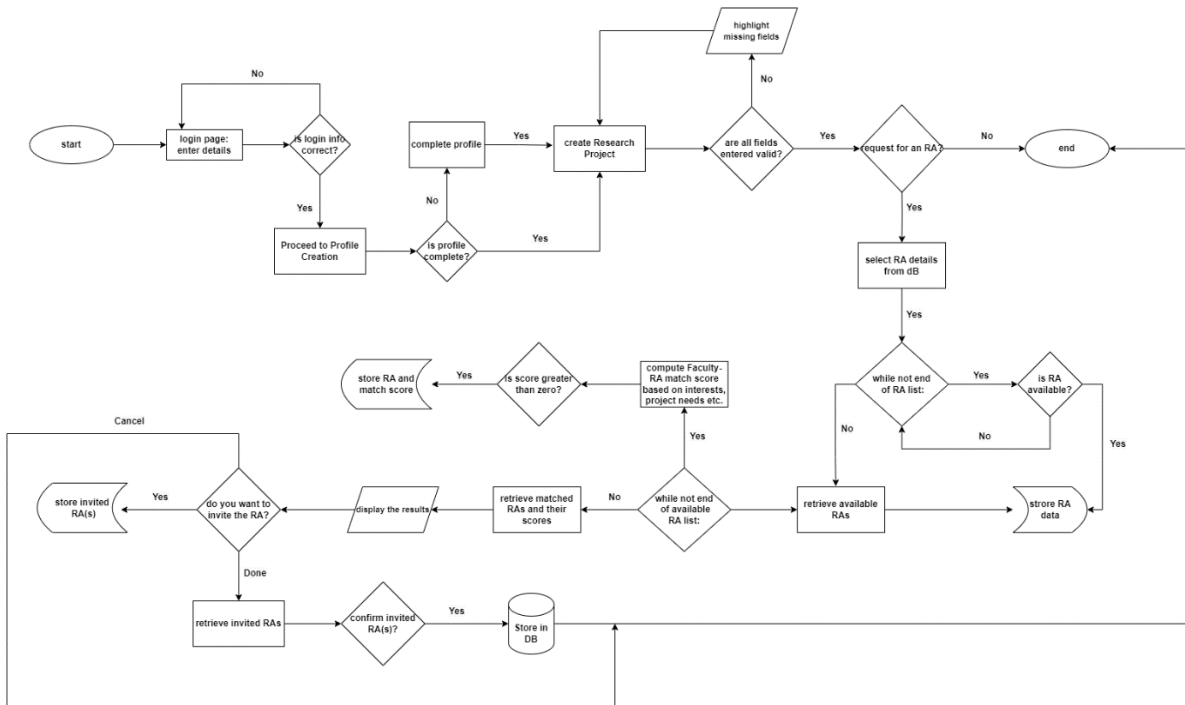
#### 1. PROCESS FLOW DIAGRAMS

Process Flow Diagram for RA profile completion and Project Request.



FLOW 1 depicts the process flow diagram for an RA profile completion and project requests.

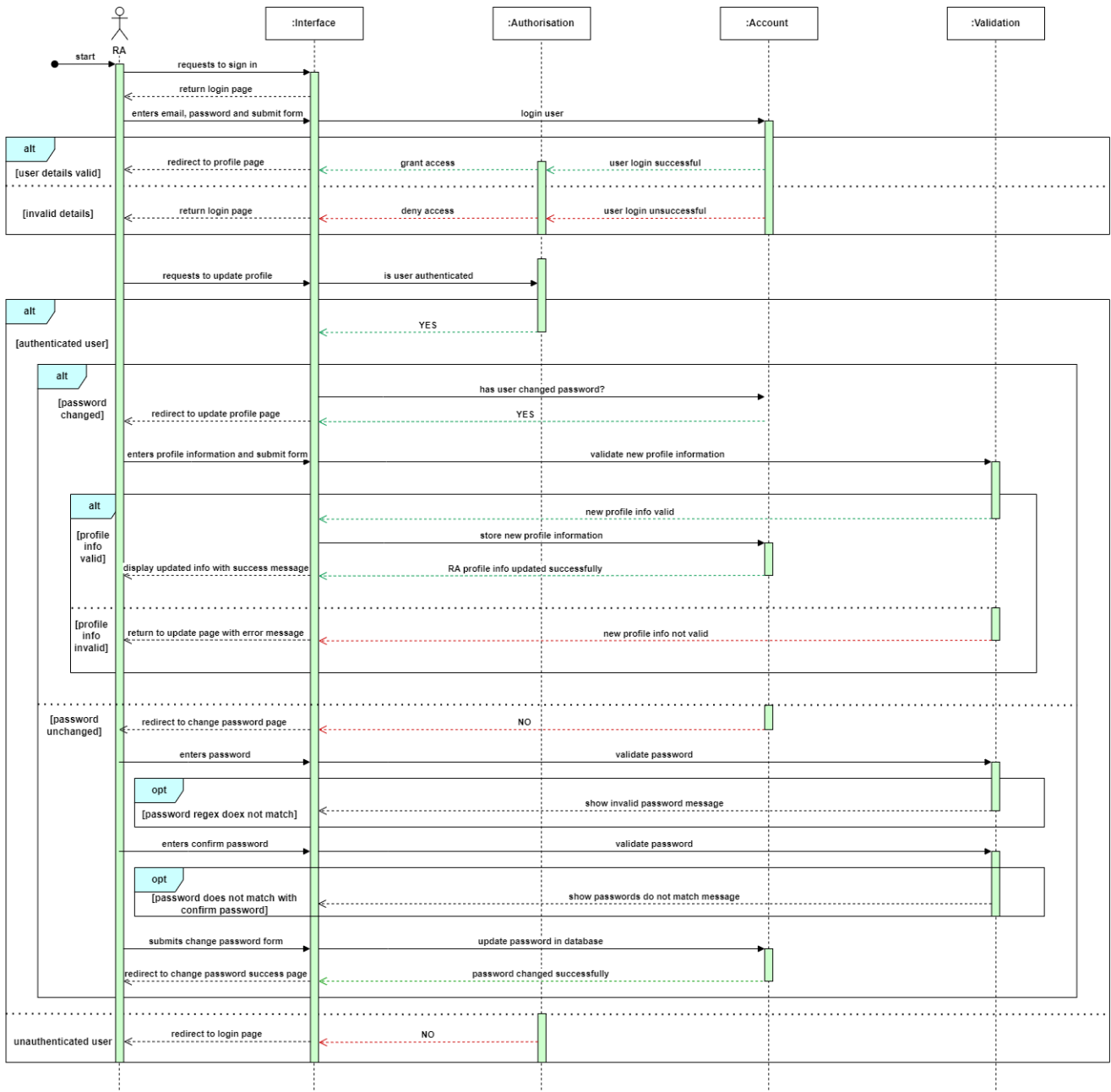
Process Flow Diagram for Faculty Profile completion, Project Creation and RA Matching.



FLOW 2 represents the process flow diagram for faculty profile completion, project creation and RA matching.

## 2. SEQUENCE DIAGRAM

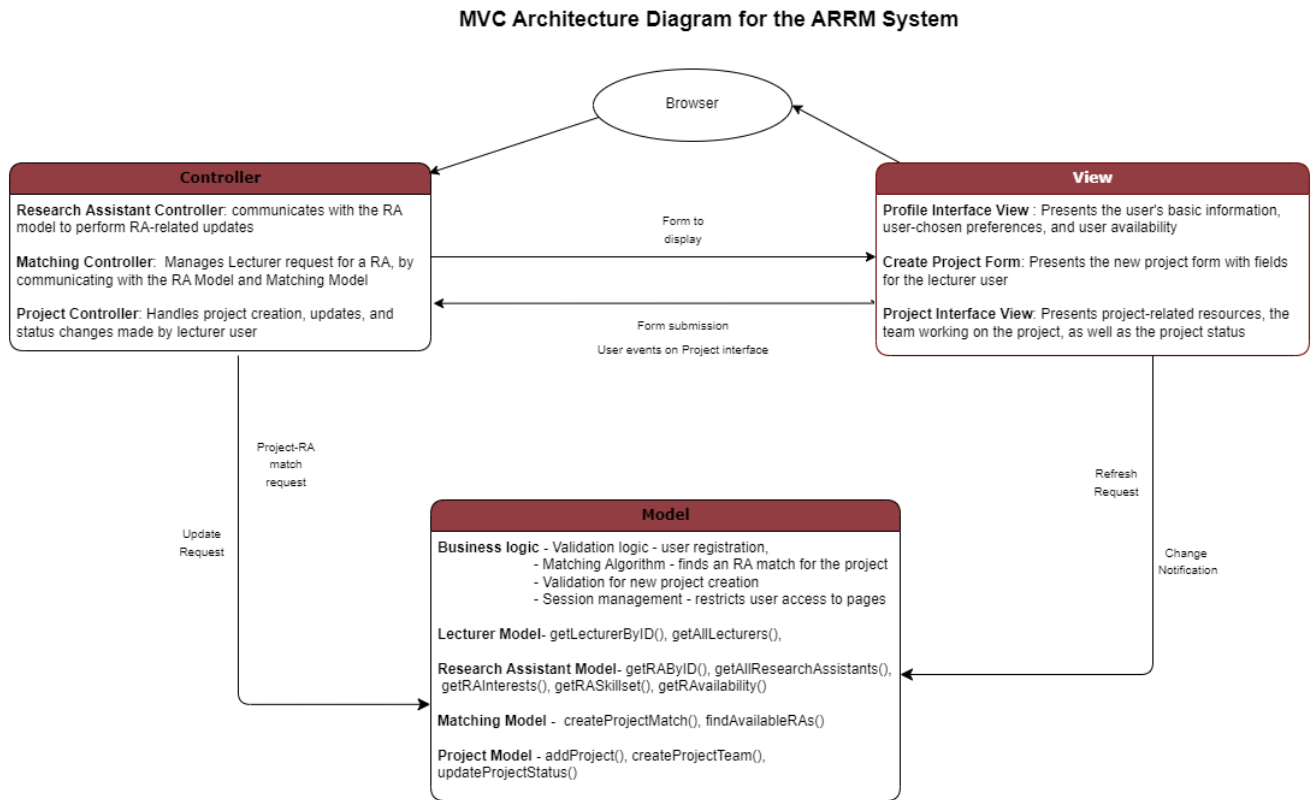
Sequence Diagram for RA Profile Completion



SEQUENCE 1 shows the sequence diagram for an RA profile completion.



### 3. ARCHITECTURE DIAGRAM



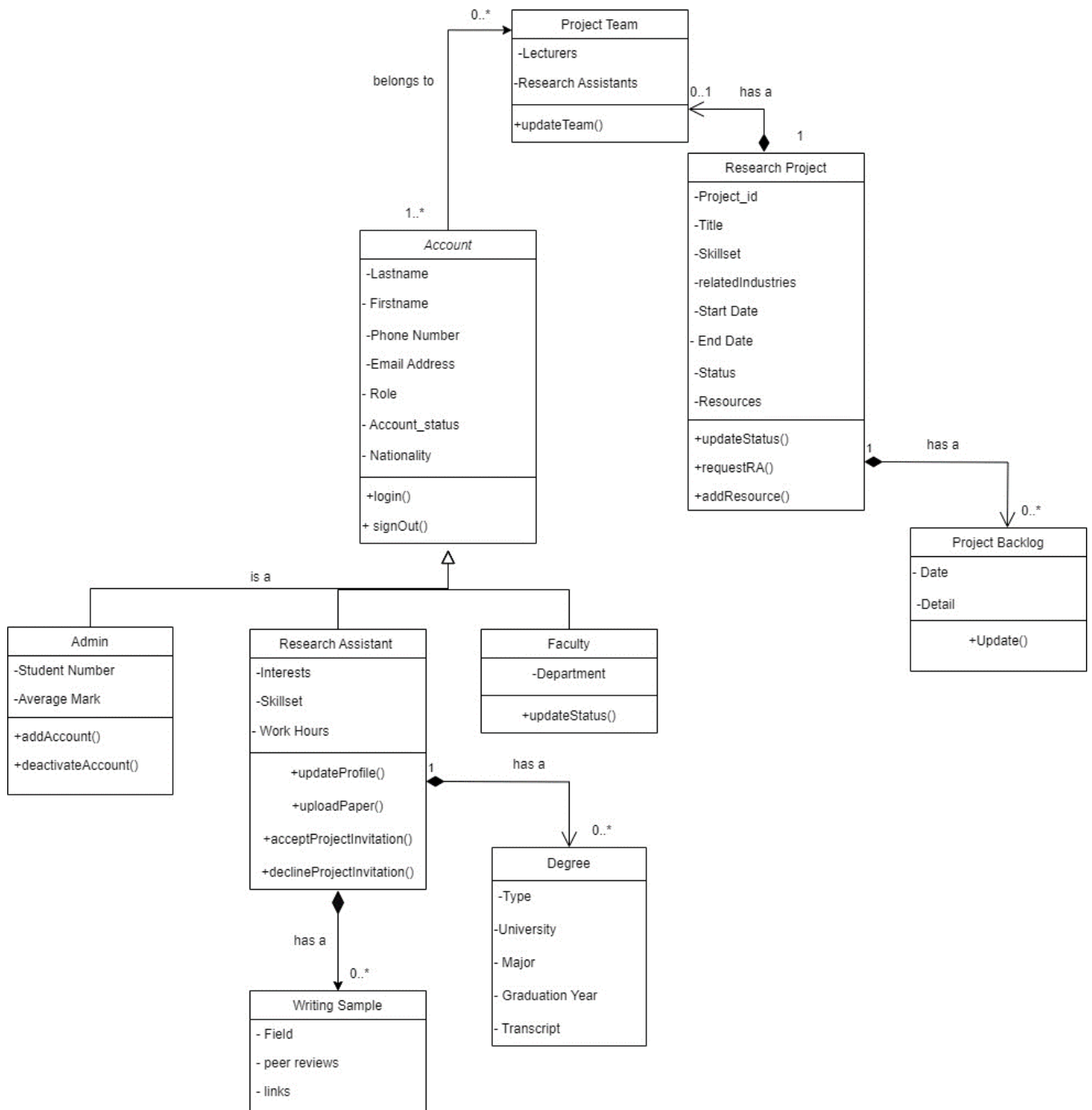
ARCHITECTURE 1 represents the system architecture diagram for the ARRM system.

	MVC For AARM Software
Description	<p>It keeps the <b>presentation</b> of the Project details, Lecturer and RA <i>profiles</i> separate from the system data.</p> <p>The <b>Model</b> component has 3 components that control the system data and <i>associated operations</i> for updating RA skillsets, project details, and profile information.</p> <p>The <b>View</b> component defines and manages how the RA, Lecturer and Project data are presented to the user.</p> <p>The <b>Controller</b> component manages user interactions such as clicking on the save button or requestRA button.</p>
Reason	<p>To build a 3-tier <b>architecture</b> with components that can be built <b>iteratively</b> and <b>independently</b>. This will allow for the users to benefit from the <i>implemented</i> functionalities while the developers make <b>incremental</b> updates to add other functionalities to the platform</p>

ARCHITECTURE 2 shows the MVC table for the ARRM system.

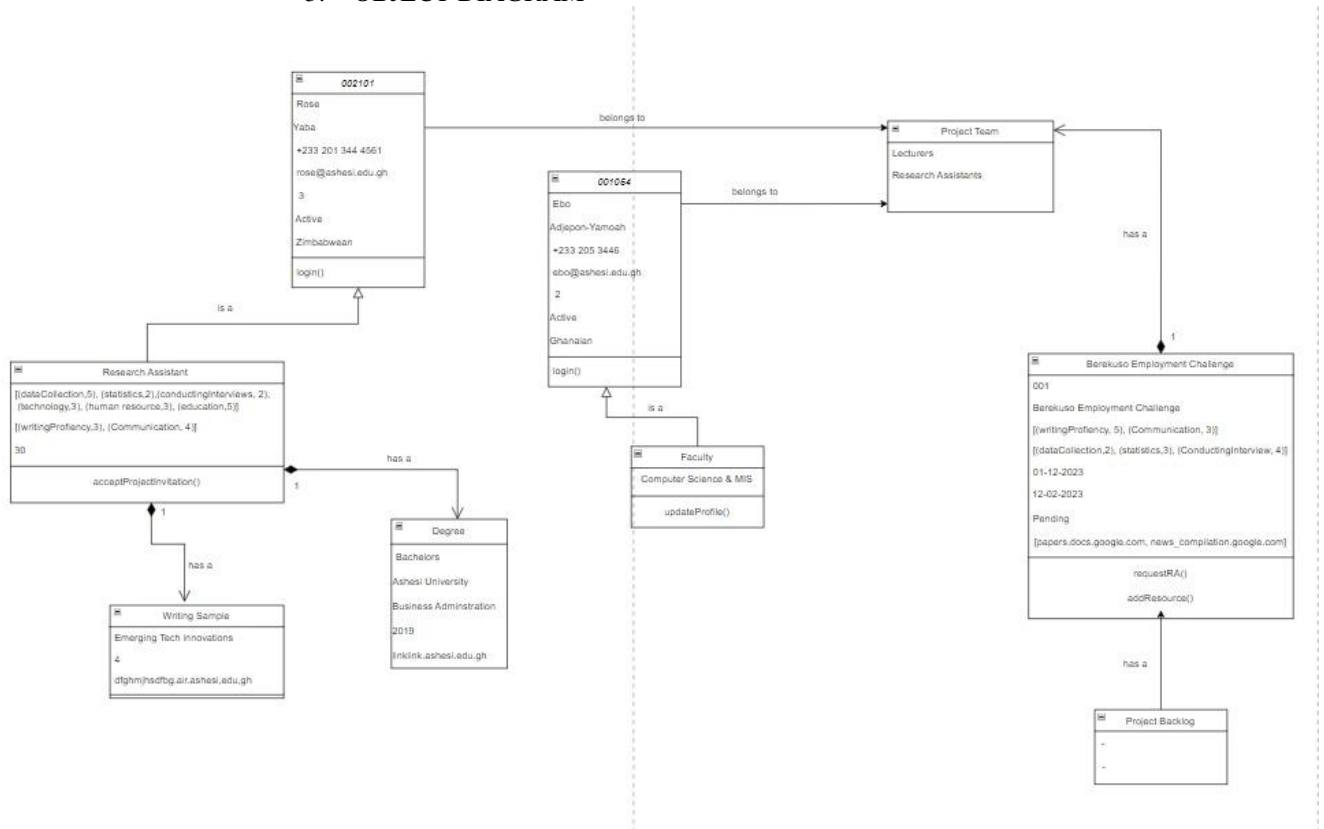


#### 4. CLASS DIAGRAM



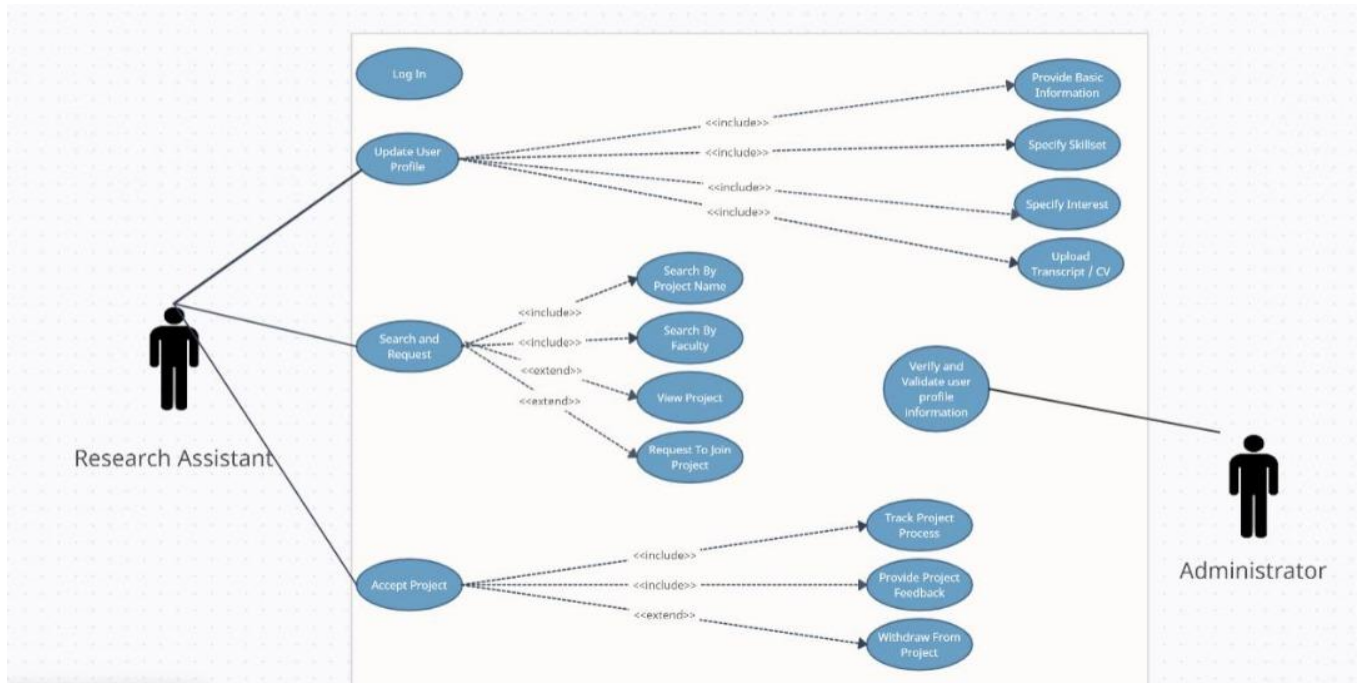
CLASS 1 depicts the class diagram for the ARRM system.

## 5. OBJECT DIAGRAM

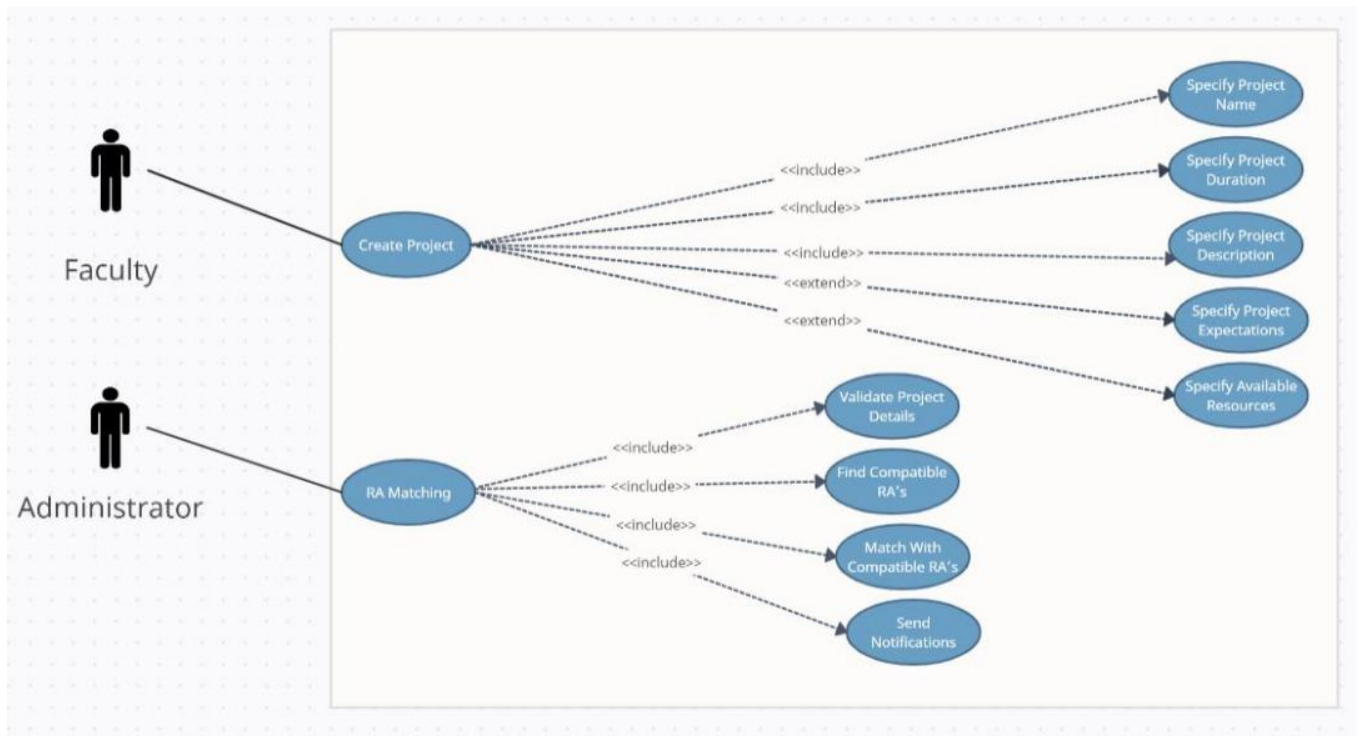


OBJECT 1 represents an object diagram for the ARRM system.

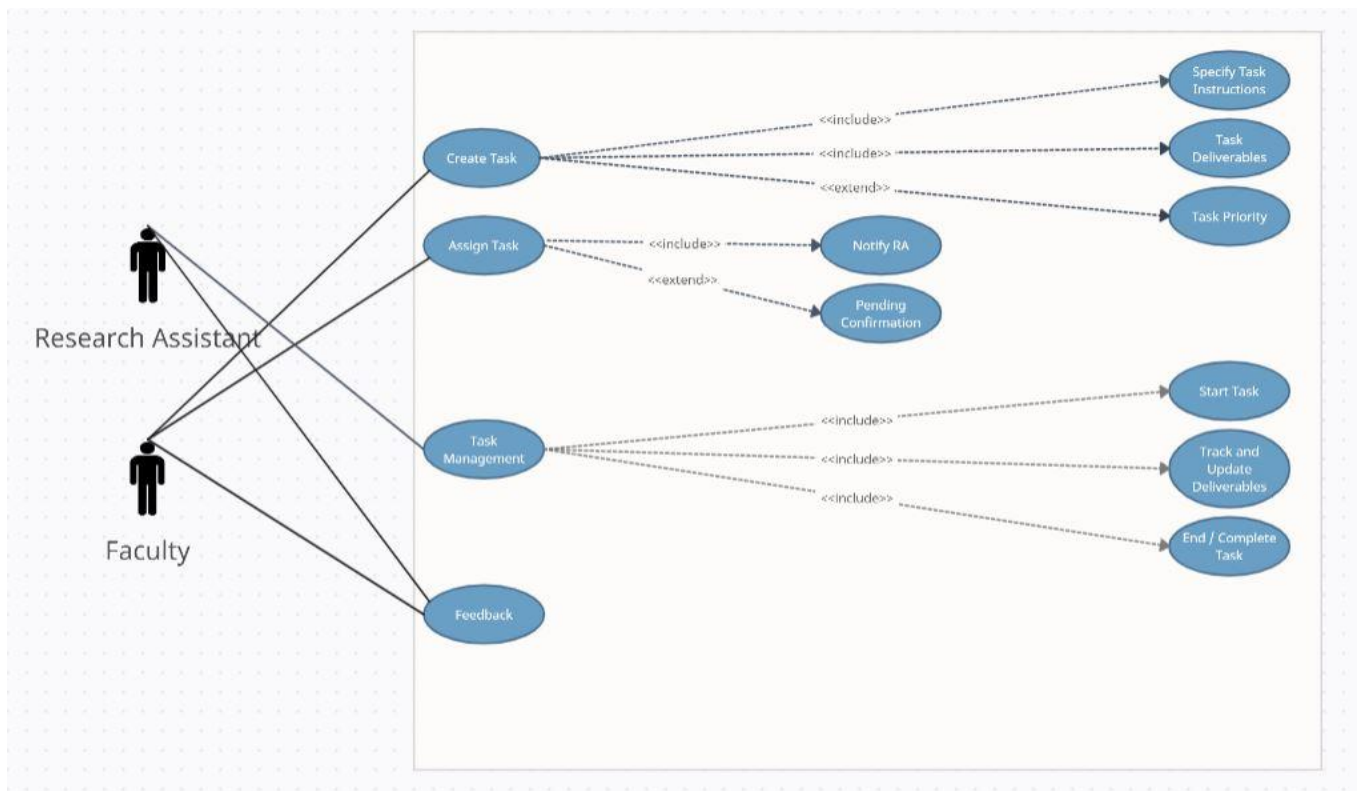
## 6. USE CASE DIAGRAM



USE CASE 1 shows the use case diagram for RA profile creation.

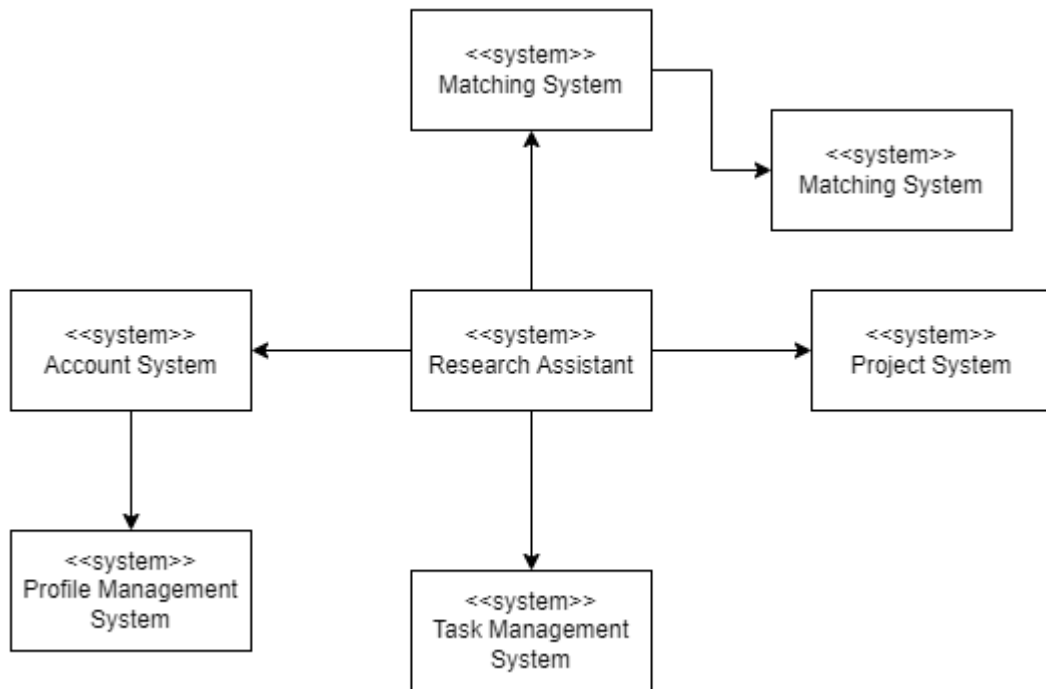


USE CASE 2 represents the use case diagram for Research Project creation and RA Matching requests.

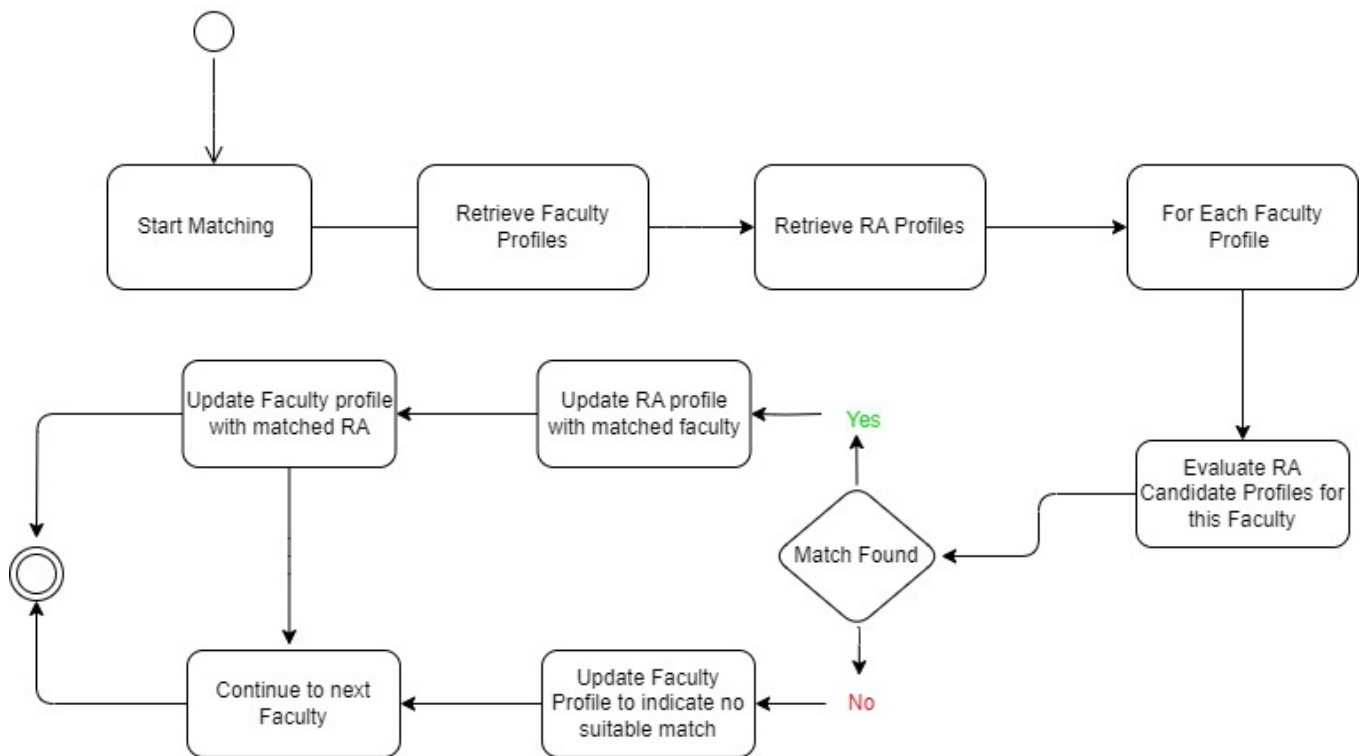


USE CASE 3 represents the use case diagram for task creation and feedback management.

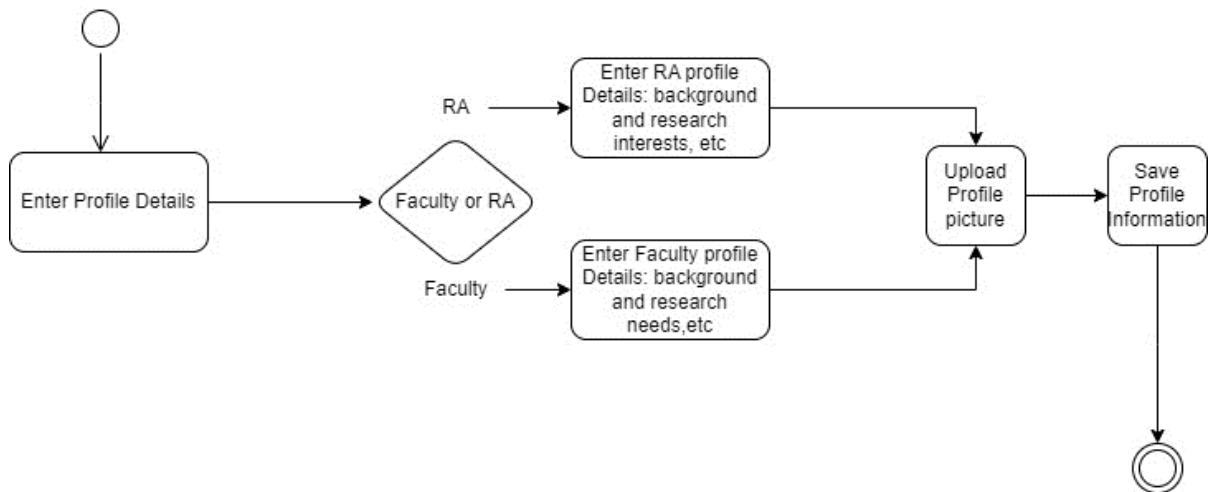
## 7. CONTEXT DIAGRAM



## 8. ACTIVITY DIAGRAM

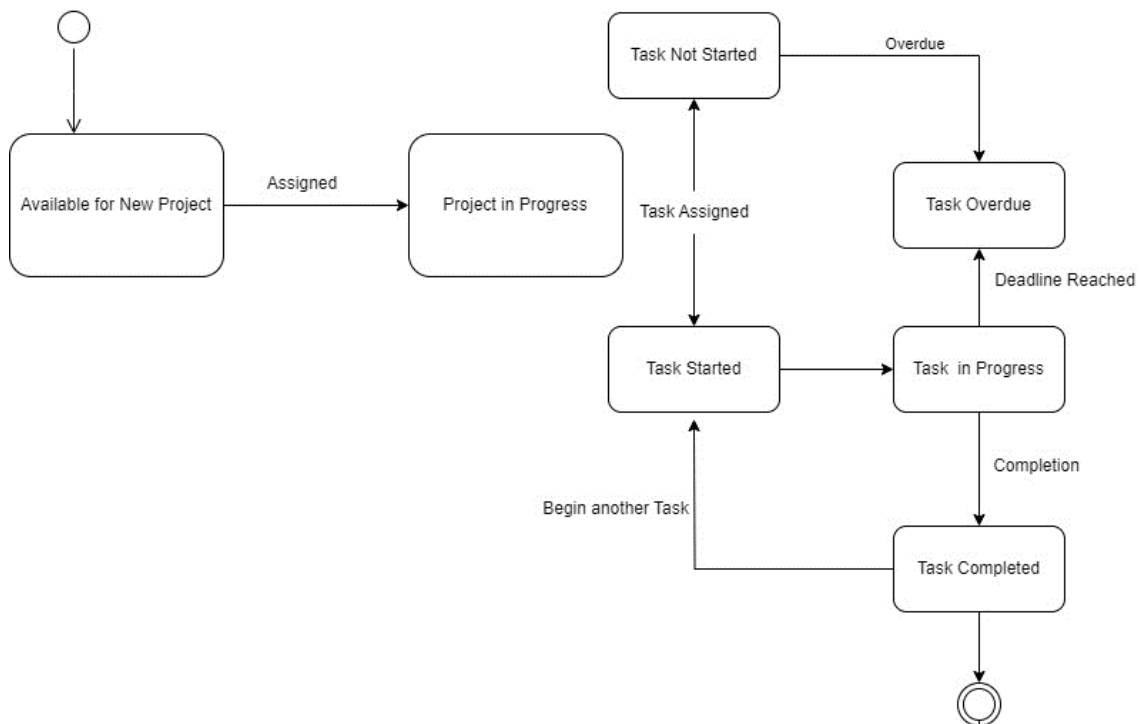


ACTIVITY 1 shows the activity diagram for RA-Faculty Matching.



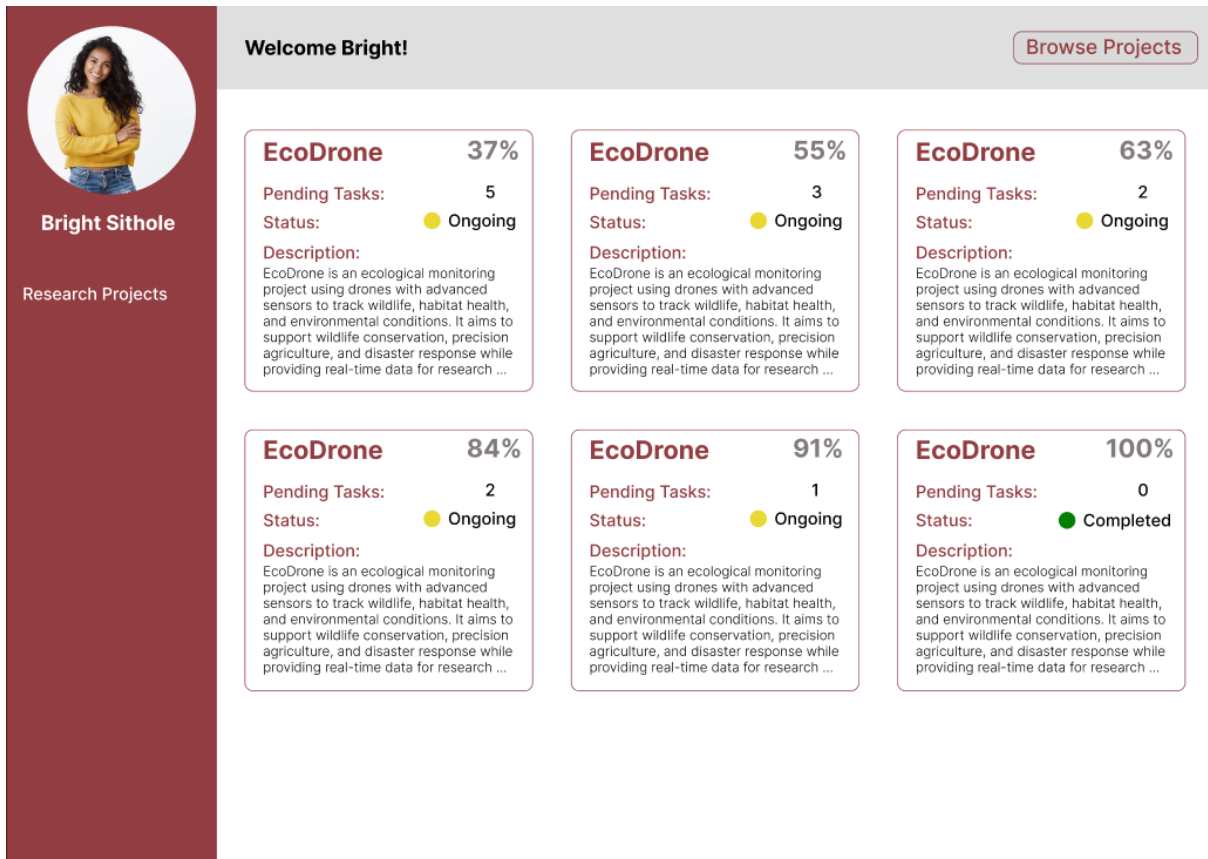
ACTIVITY 2 depicts the activity diagram for RA and Faculty profile creation.

## 9. STATE DIAGRAM



STATE 1 represents the state diagram for the ARRM system.

## B. FIGMA DESIGN



FIGMA 1 represents the [FIGMA](#) design for an RA dashboard on the ARRM system.