**CSc 179**

**Software Testing and Quality Assurance**

**Summer 2022**

**Software Req and Design Brief V1.2**

**Project Deliverable # 2**

#### Due Date: End of Term

The **Software Req and Design Brief purpose is to document project scope, software requirements, models. Design and project assumption and constraints. This is meant to be a living document so teams should continuously improve and update its contents.**

The **Software Req and Design Brief** should include table of content page and a title page – the title page should have the following - Course title, course section, team name, team members, project name, and deliverable # 2.

***Grading Criteria:***

* ***Completeness (all sections are complete and well documented)***
* ***Accuracy of content and models***
* ***Quality of documentation***
* ***Completeness and quality of the delivered product/system***

**Software Req and Design Brief –V1.2**

1. **Software Engineering Team**

**Team 0.5**

Tim Huang

Ryan Farruggia

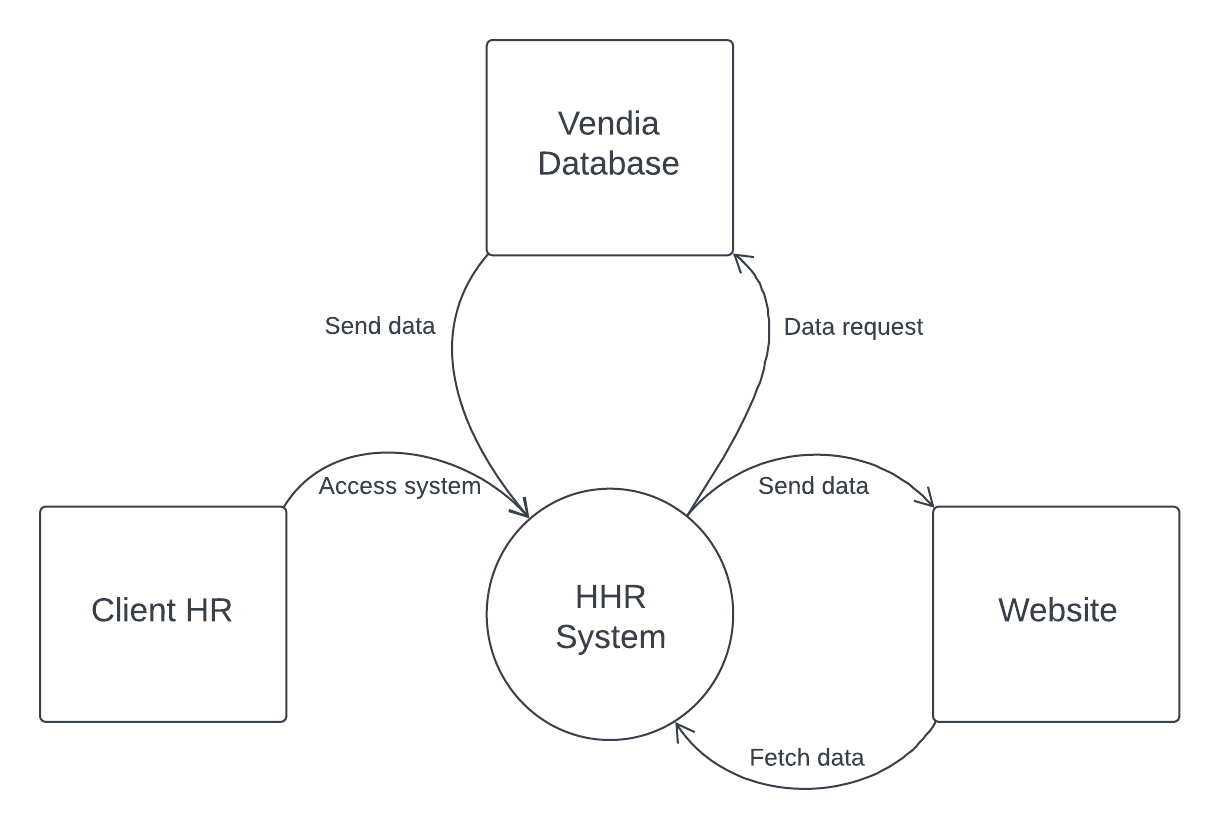
Joshua Degmetich

Kenneth Munk

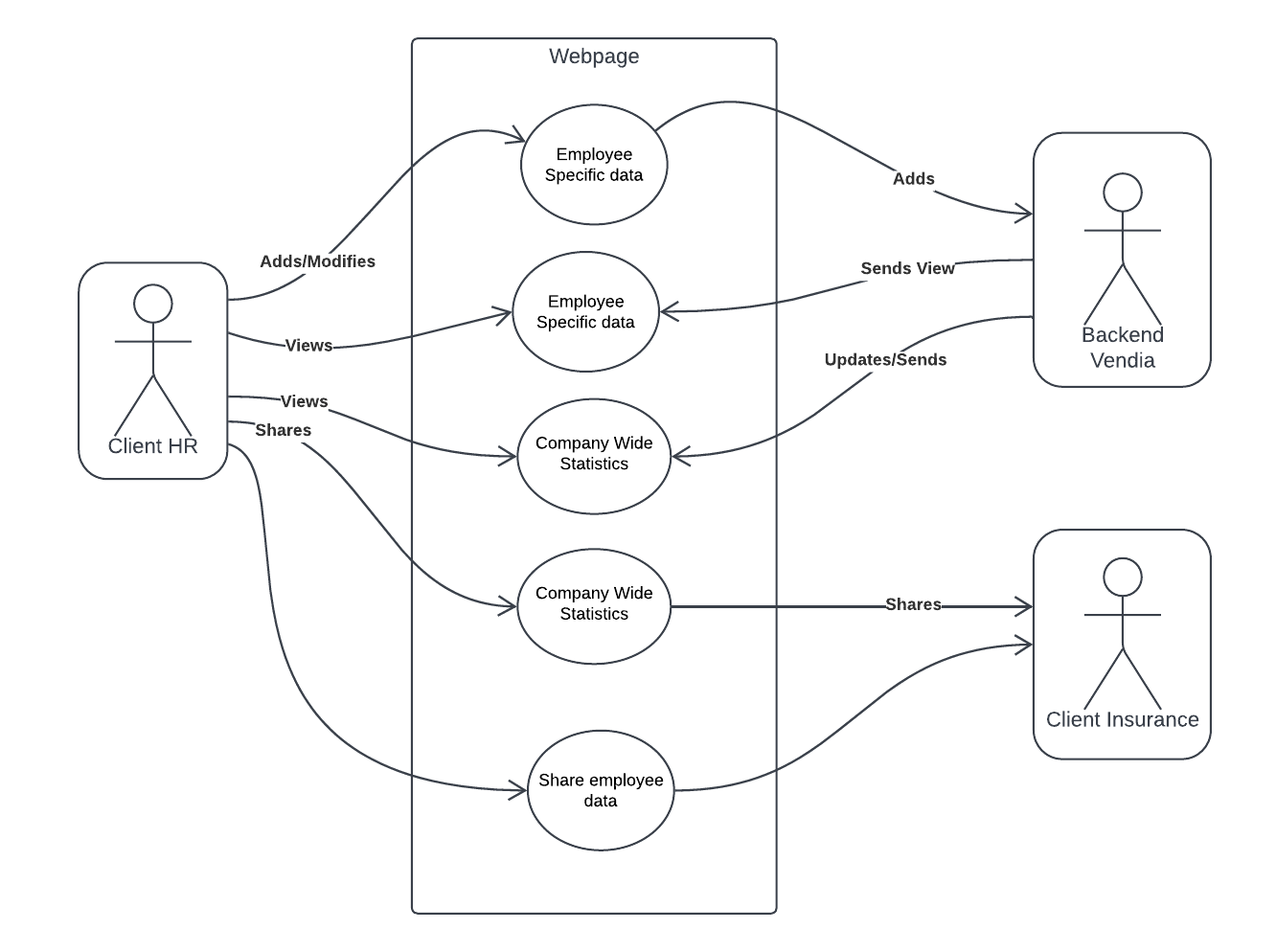
Sean Hackett

1. **Project Scope**

Create a System Context Diagram (SCD) – add a brief description of SCD



1. **Requirements discovery and analysis; choose either approach 3.1 or 3.2**

* **Using Approach 3.1**
  + **Use Case Model**
    - Diagram  
      
    - At a high-level view, the use cases for the company health page for HR related clients is to add employees for tracking, view employee specific data and view company-wide statistics. The other uses that it will have will be to forward specific employee data and/or company-wide data to insurance companies in order to shop for better group policy deals
  + Testing
    - Test Case Criteria
    - Product Testing Environment
    - Testing logistics assignments
    - Test Cases based on use case model
    - Documentation of test cases and test case results (see test report)
  1. **Object Oriented Requirements Analysis (OOA) – UML Modeling**
* **Use Case Model**
* Refer to previous section “Requirements” for UML diagram.
* Most typical use cases will flow through client views, adds/modifications, and views. Less common use cases will include sharing information with client insurance companies. Though this information is automatically entered into Vendia’s database, it is not necessarily automatically shared with insurance companies. This means that sharing requires some access control depending on who requests it.
* **Testing:**
* Clearly define test case criteria
* Create a product testing environment using supported platforms
* Assign testing logistics to appropriate team members based on their roles
* Design the test cases based on the Use Case model (Use case and Use Case’s ‘flows
* Document all test cases and relevant test results –\* see the test report

1. **System Requirements**

**5.1 FRs**- Based on the project description and the Use Case model, list all system functional requirements.

*Number the Functional Requirements (FR1, FR2, FR3, etc.) in a systematic manner****.***

* + ***FR1 – HR client needs to be able to add employees to the tool for tracking***
  + ***FR2 – HR client needs to be able view company wide statistics on dashboard***
  + ***FR3 – HR client needs to be able to look up***
  + ***FR4 – HR client needs to be able to be able to view employee specific stats***
  + ***FR5 – HR client needs to be able to share employee specific data with insurance companies in order to shop for employee rates***
  + ***FR6 – HR client needs to be able to share company-wide statistics with insurance companies in order to shop for group policy rates***
  + ***FR7 – Third-party Insurance companies must be able to view company wide statistics***
  + ***FR8 – Third-party insurance companies must be able to view only the employee specific data and not data that was not shared with them.***
  + ***FR9 – Third-party insurance companies must be able to view company-wide statistics that are shared with them***
  + ***FR10 – Third-party insurance companies that have not had company-wide statistics shared with them should not be able to view said data***
  + ***FR11 – HR Client can only view employee data of employees within the organization that they are in responsibility for***

**5.2 NFRs -** system attributes such as usability, reliability, and performance, etc.

* + **Capacity** – Dashboard should allow at least 10 employees to be added to its database
  + **Reliability** – Dashboard should have the ability to output accurate calculations of cross-sectional data if implemented
  + **Security** – Dashboard should only be operable by HR employees and data that is shared does not reveal personal information
  + **Performance** – Dashboard should not have unreasonable load times
  + **Availability** – Employers can view and use the dashboard throughout the week at any time during the workday
  + **Usability** – Easy to utilize the dashboard’s functions
  + **Compatibility –** Dashboard should be able to operate on desktop systems
  + **Scalability** – Dashboard should allow for the addition of more employees along with different types of calculations and measurements to determine health insurance plans
  + **Data Integrity** – All employee data is input accurately into the dashboard by HR employees only
  + **Portability** – Dashboard should be operable on most web browsers or operating systems on desktop systems
  + **Modifiability –** Allow for the ability to change which metrics are used to calculate health insurance plans

1. **Data Design**

* Develop an ERD diagram - Briefly describe the ERD

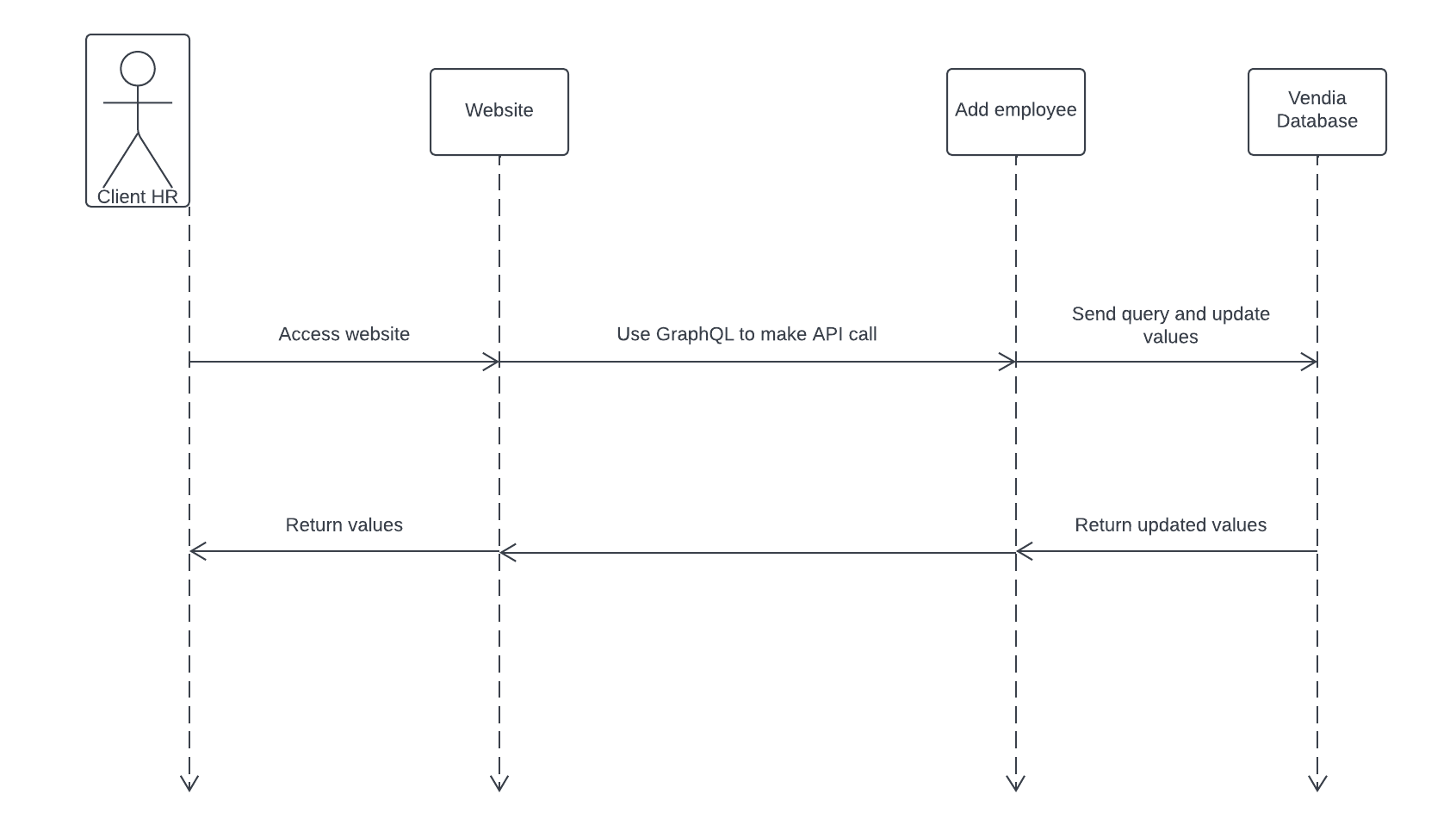
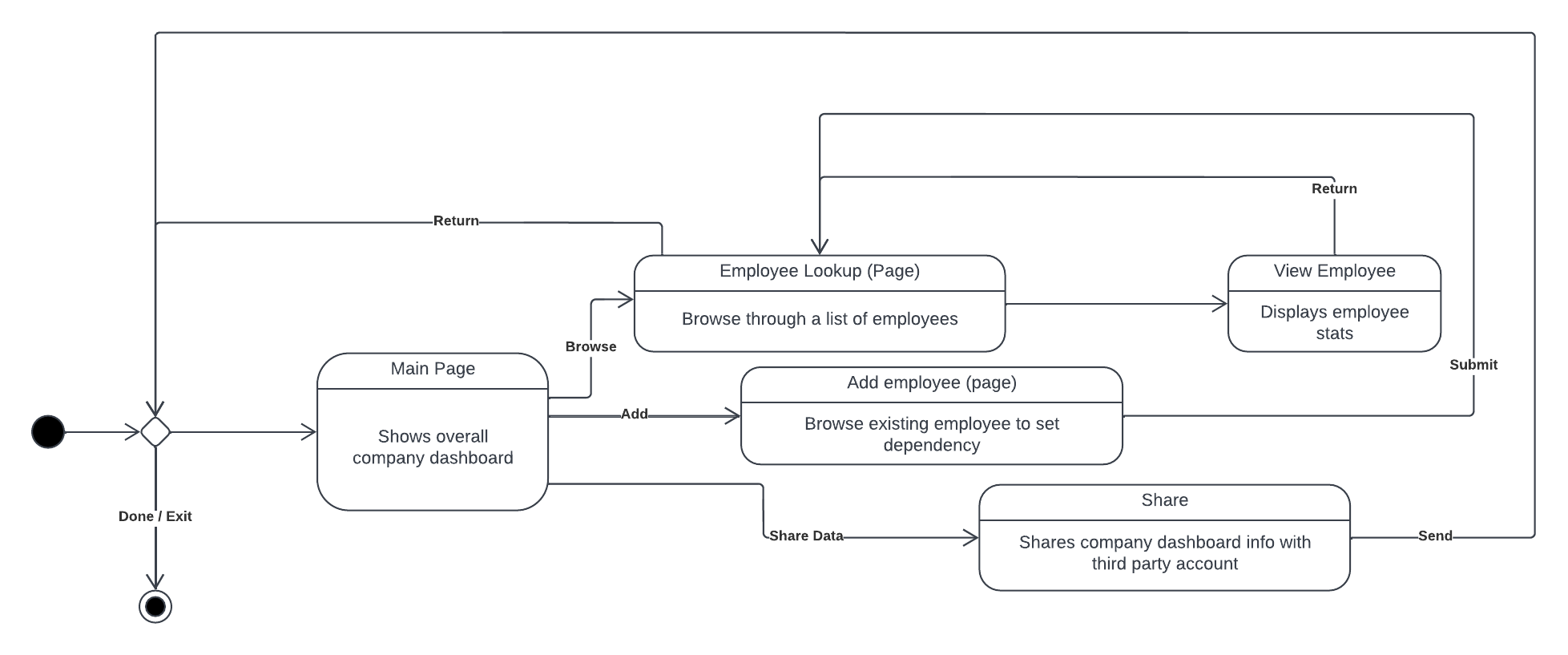
The Data Design describes structures that reside within the software. Attributes and relationships between data objects dictate the choice of data structures

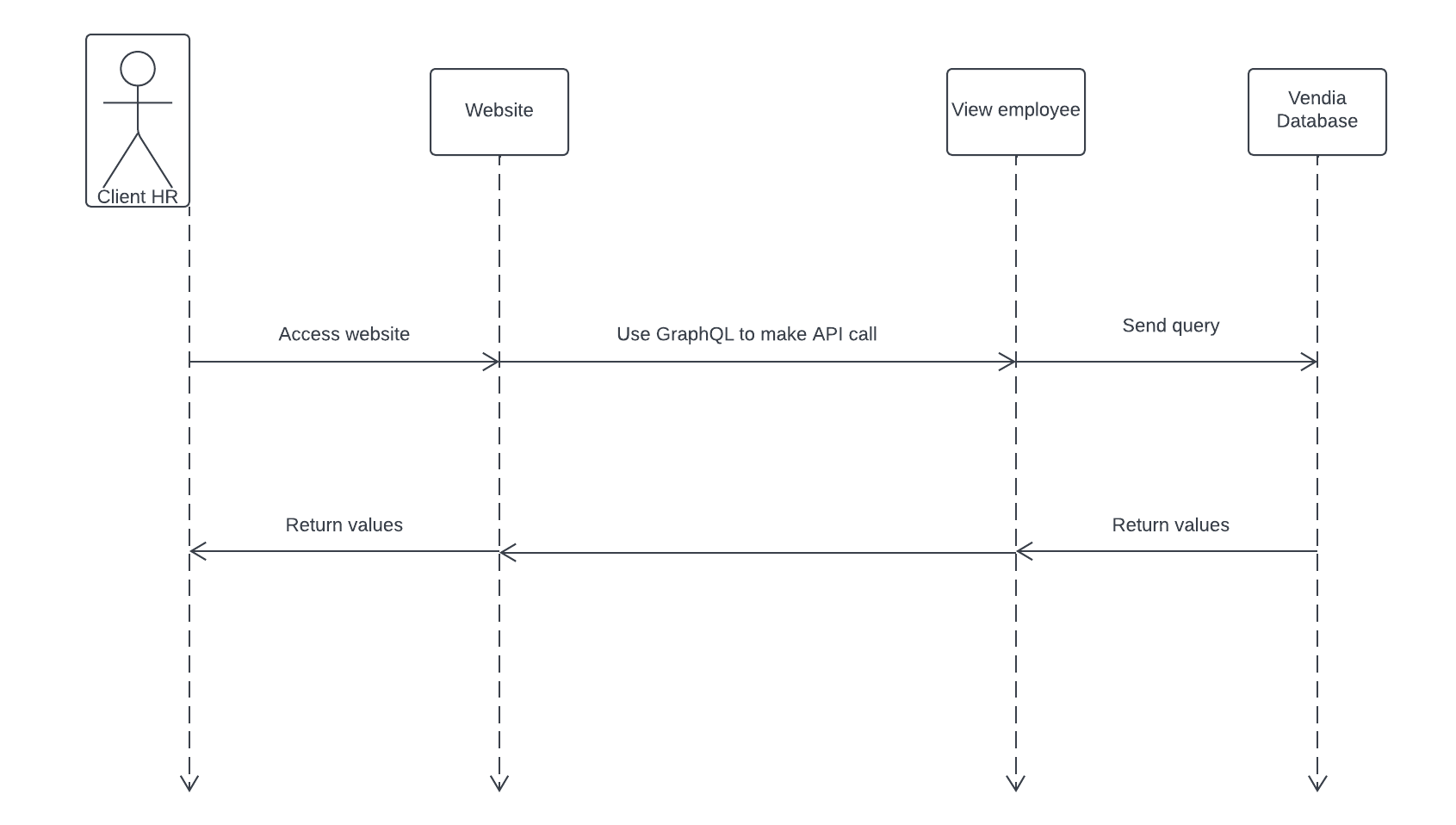
Diagram

Description automatically generated

1. **Detailed Design**

* Create a set of interaction models (i.e. sequence diagrams) to capture low level design view of the system
* Interaction Diagram

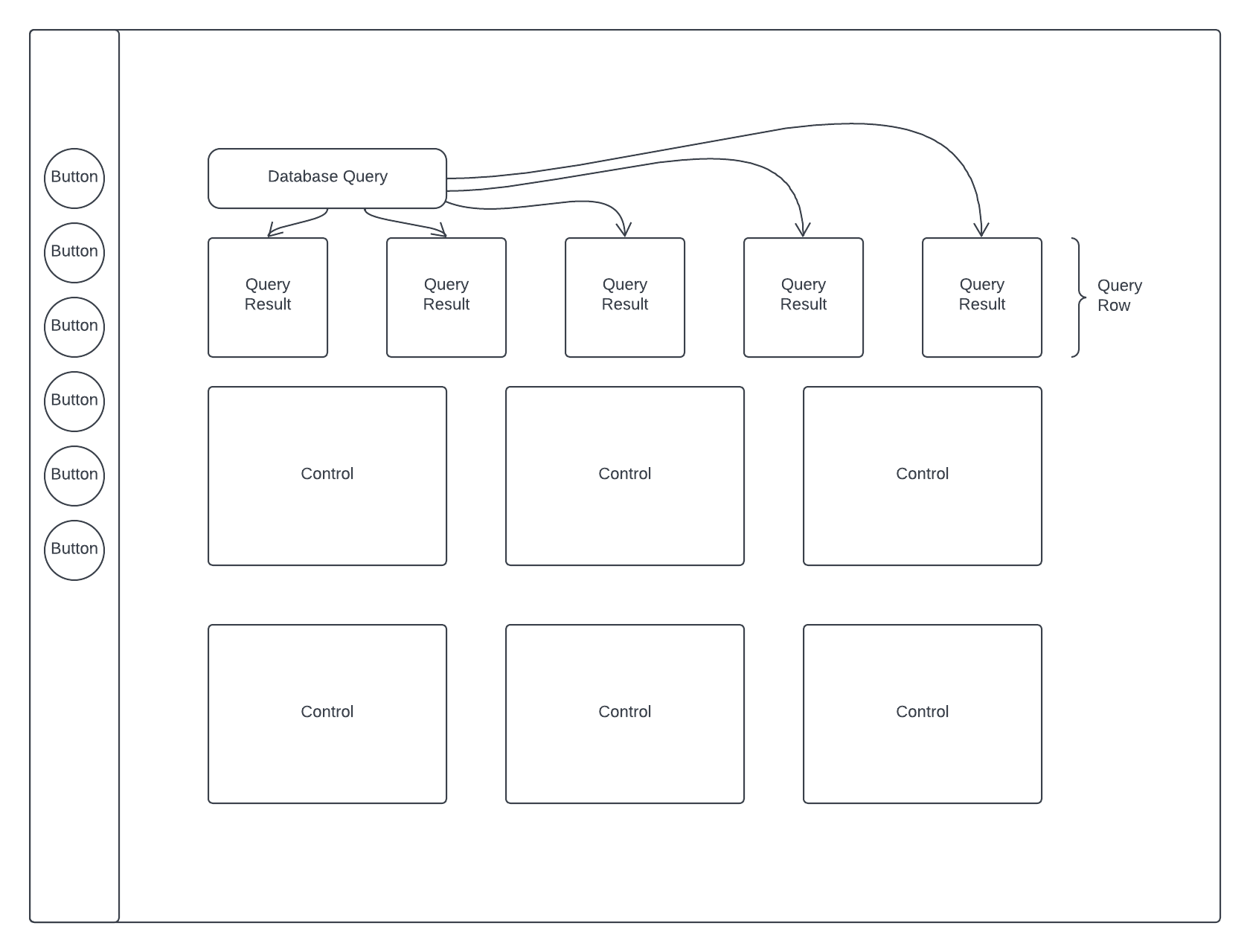




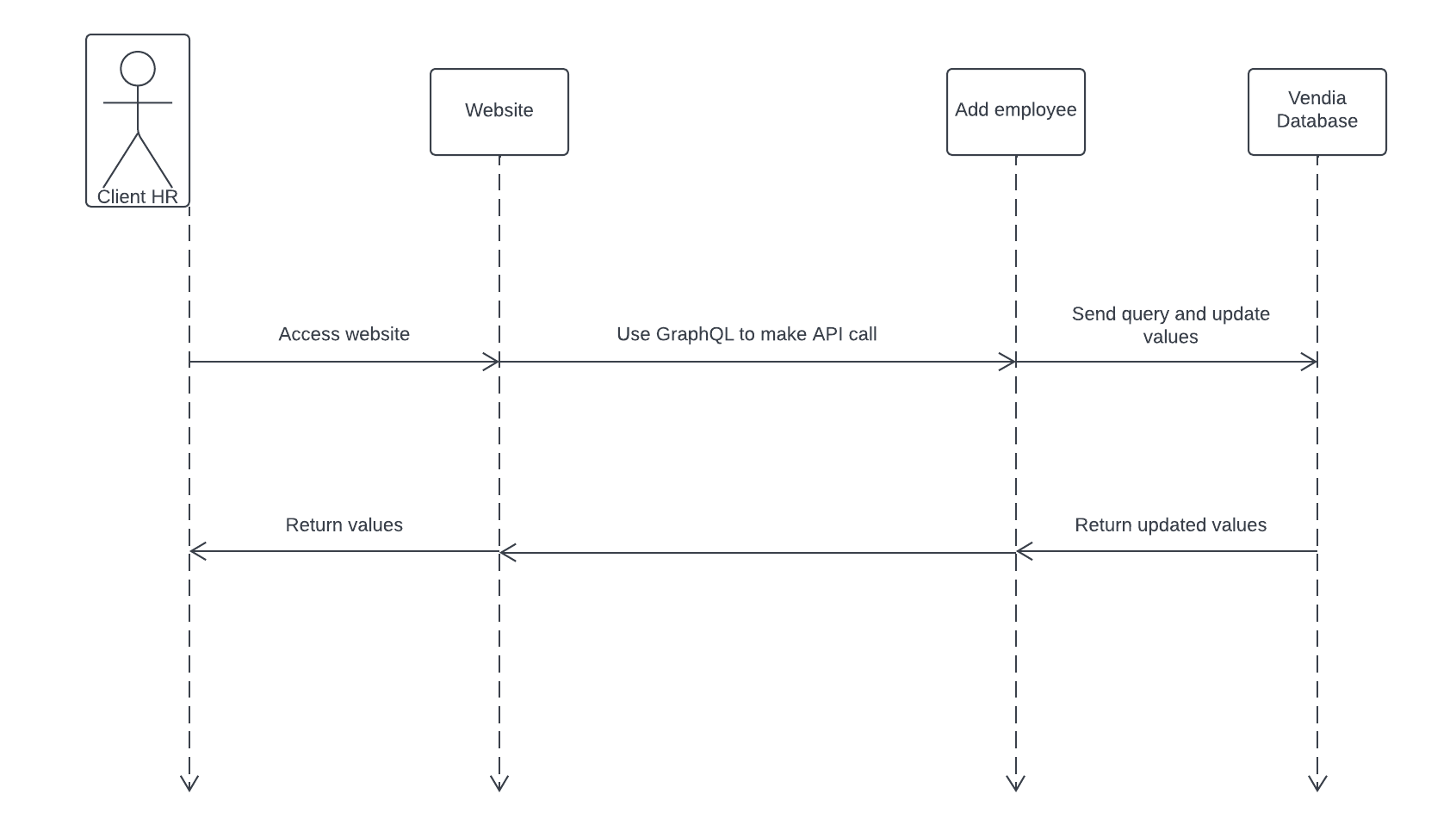
1. **User Interface Design**

The Interface Design describes internal and external program interfaces. Interface designs are based on the information obtained from the analysis models. Use Cases, User Stories, and Sequence diagram to capture the interface design.

**External program interface:**



**Internal program interface:**



1. **Technology and Tools**

* Front end: HTML, Sass, JavaScript
* Backend: GraphQL (Vendia API)
* Database: Vendia
* Version control: GitHub

1. **Assumption and constraints**

Any relevant assumptions and any special design issues, which impact the design or implementation of the software, are noted here

Assumptions:

* + Assume only HR employees can access the dashboard (if authentication not implemented)
  + Assume all data input is accurate
  + Assume no second WebApp is needed to display Third Party Client’s data received

Constraints:

* + Backend technology used for this project is Vendia
  + Data shared to Third Party Client’s Vendia account can only view data without personal identification
  + Dashboard contains a minimum of 10 employees with personal data and metrics