

Report on
AllyHub

Prepared for

Dr. Taslim Taher Assistant Professor Department of CSE	Ms. Tasnuva Binte Rahman Lecturer Department of CSE
---	--

Course No: CSE 3224

Course Name: Information System Design & Software Engineering Lab

Prepared by

Lab Section: A2

Group Name/No: A202

ID: 20210104032 Name: Afia Fahmida
ID: 20210104040 Name: Ashikul Islam
ID: 20210104047 Name: Zenun Chowdhury

Date: 26 June, 2024



Department of Computer Science and Engineering

List of Activities:

For Clients:

- ✓ Clients can sign up and login
- ✓ Clients can post and edit project proposals
- ✓ Clients can browse, search, filter out developers list
- ✓ Clients can send collaboration requests to Developers
- ✓ Clients can view, accept, decline applications sent by developers
- ✓ Clients can make payment
- ✓ Clients can give and get feedbacks

For Developers:

- ✓ Developers can sign up and login
- ✓ Developers can browse, search, filter project proposals
- ✓ Developers can apply for projects
- ✓ Developers can accept or decline collaboration request
- ✓ Developers can browse, search, filter out developer list
- ✓ Developers can receive payment receipt
- ✓ Developers can give and get feedbacks
- ✓ Another point to be noted, a potential developer has access to all the actions a client can do which means at certain time a developer himself/herself can turn into a client.

Main Process and Sub-process Names:

1. Process User Data

- 1.1. Sign up
- 1.2. Login

2. Project Management

- 1.1. Post Project proposal
- 1.2. Edit Project proposal
- 1.3. Browse projects
- 1.4. Search projects
- 1.5. Filter projects
- 1.6. View projects

3. Interaction Management

- 1.1. Send collaboration request
- 1.2. Accept or decline collaboration request
- 1.3. Apply for projects
- 1.4. Accept or decline application
- 1.5. Make payment
- 1.6. Get Receipt

4. Review Management

- 1.1. Give reviews
- 1.2. Get reviews

5. Exploration Management

- 1.1. Browse Developers list

- 1.2. Search Developers list
- 1.3. Filter Developers list

Context Level Diagram:



This diagram shows the basic connection between the entities and process of our project AllyHub. It also displays the activities using arrows from the very surface level. Here,

Entities: Clients, Developers

Process: AllyHub: An interactive platform for Developers and Clients

Activities of Clients:

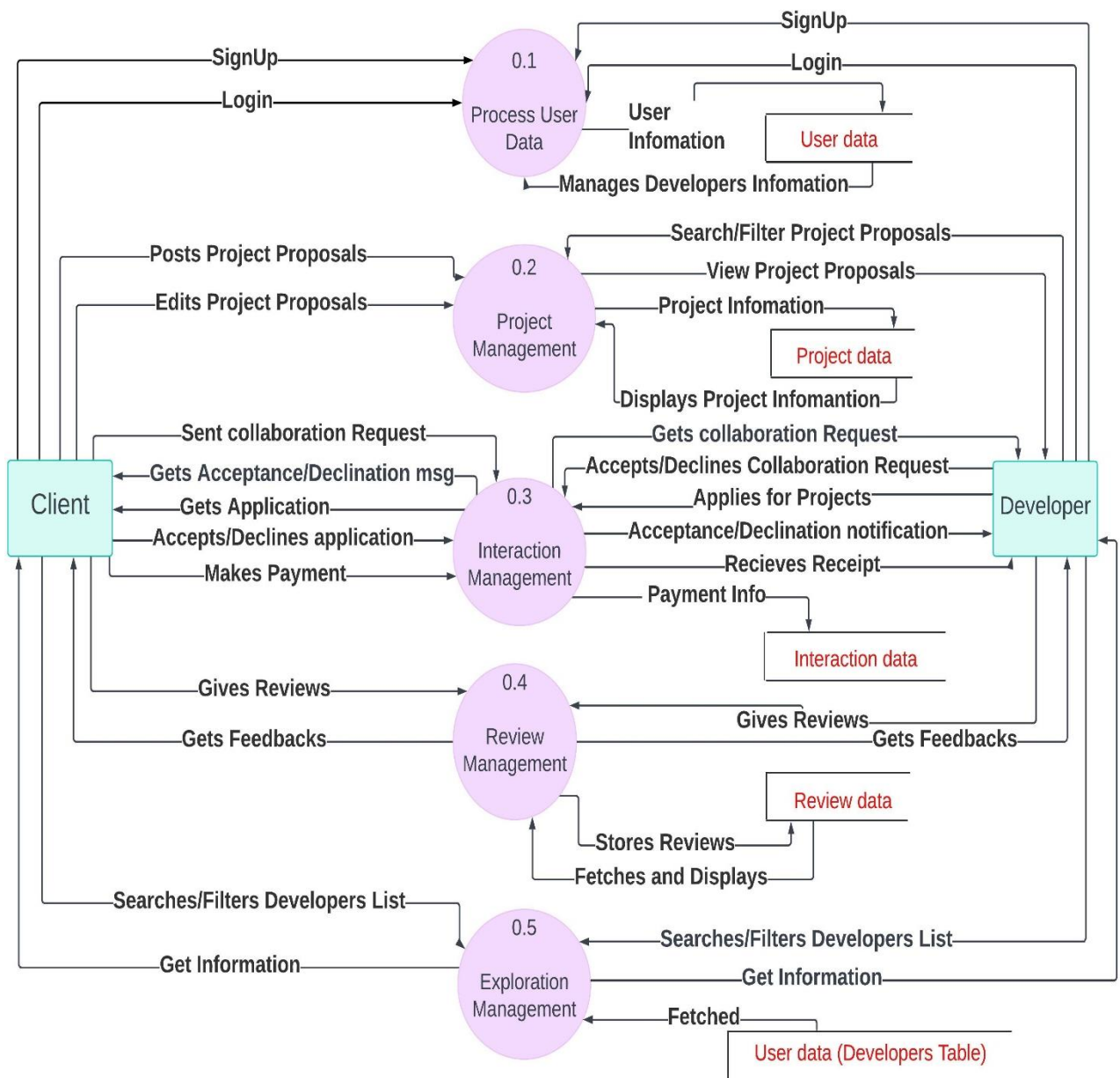
1. Clients need to sign up first and get confirmation that it is done properly.
2. Clients can log into their individual account.
3. Clients can browse and search up filtering developers list and view them.
4. Clients can post project proposals.
5. Clients can receive applications from developers and accept or decline thereby.
6. Clients can send collaboration requests that in our platform we have claimed as "Handshake" feature.
7. Clients can make payments once the assigned task of the developers are done through the payment gateway method.
8. Clients can give and get feedbacks.

Activities of Developers:

1. Developers need to sign up first and get confirmation that it is done properly.
2. Developers can log into their own accounts to continue.
3. Developers can also browse and search up filtering developers list and view them tom team up.

4. Developers can browse and search up for projects filtering them based on topics, languages etc.
5. Developers can apply for projects.
6. Developers will get notifications whether the client accepted or declined their application.
7. Developers can accept or decline “Handshake” requests.
8. Developers will receive the payment receipts once they end their task.
9. Developers can give and get feedbacks.

Level-0 Diagram:



This Diagram breaks down and shows the main processes more widely with precise titles to give a better view on all the actions the two entities of AllyHub are associated with and the data flow within the system. Here,

Entities: Clients, Developers

Process:

1. Process User data
2. Project management
3. Interaction management
4. Review management
5. Exploration management

Activities of Clients:

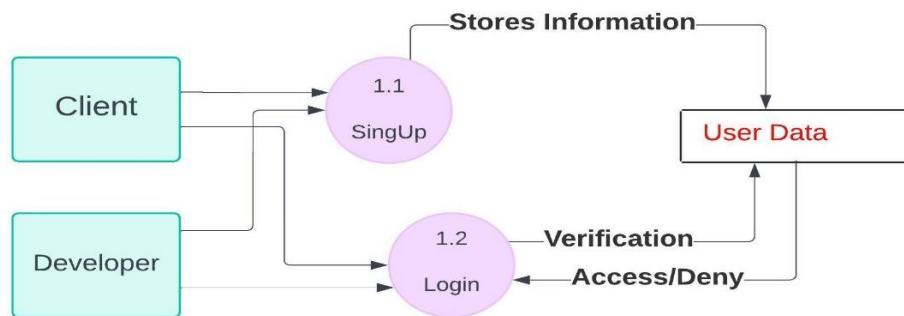
1. Process User data: Clients need to sign up first to store their information that will help them interact actively with other users. They can login afterwards using the data they provided that will be verified from the database user data information.
2. Project management: Clients can post and edit their project proposals which will be stored and updated respectively in the database.
3. Interaction management: Clients with their respective accounts can send collaboration requests to the developers and also go through the applications provided by willing developers to collaborate on their posted projects. They can response with acceptance or declination. Clients can also make payment once the deal of a project ends and developers finish the task they were assigned to.
4. Review management: Clients can give reviews and see the feedbacks and work experience shared by other individuals of their projects.
5. Exploration management: Clients can browse and look up for developers they need for their projects by filtrating the provided developers list according to their preferences.

Activities of Developers:

1. Process User data: Developers need to sign up first to store their information that will help them interact actively with other users. They can login afterwards using the data they provided that will be verified from the database user data information. Also their data will be used to display the developers list on frontend.
2. Project management: Developers can browse and search up project proposals filtrating them. Again a potential and skillful developer can behave as client at certain time and can post and edit their project proposals like a client.
3. Interaction management: Developers with their respective accounts can receive collaboration requests and accept or decline it. They can also apply for collaborating on the posted projects. They will get responses of acceptance or declination. Developers will receive payment receipts once the deal of a project ends and they finish the task they were assigned to do.
4. Review management: Developers can give reviews and see the feedbacks and work experience shared by other individuals of their projects.
5. Exploration management: Developers can browse and look up for developers too so they can build a potential team to work on they need for their projects by filtrating the provided developers list according to their preferences.

Level-1 Diagrams:

1. Process User data:



Entities: Clients, Developers

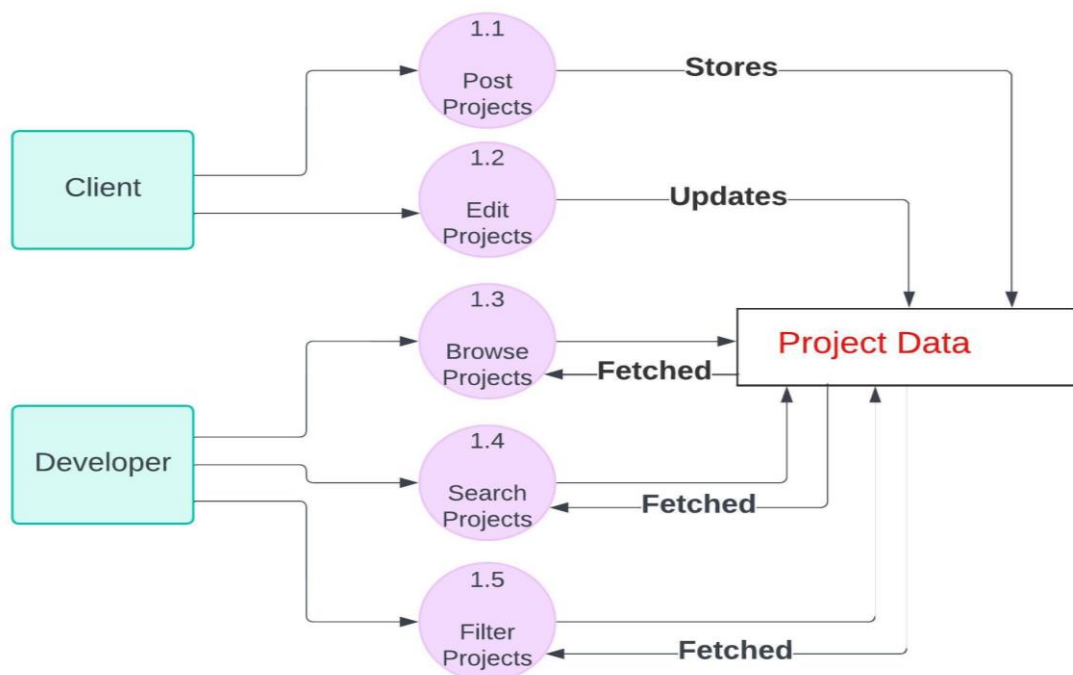
Process:

- 1.1. Sign up
- 1.2. Login

Activities:

- ✓ Clients and Developers both need to sign up first according to their role which process will store all the necessary information in user data section of the database.
- ✓ Here clients will stay clients but developers can turn into clients depending on their work experience, skillset etc.
- ✓ Both the clients and developers need to log into their accounts to get access to other features. Login process at first verifies the provided credentials from database and then notifies if account is accessed or access is denied.

2. Project management:



Entities: Clients, Developers

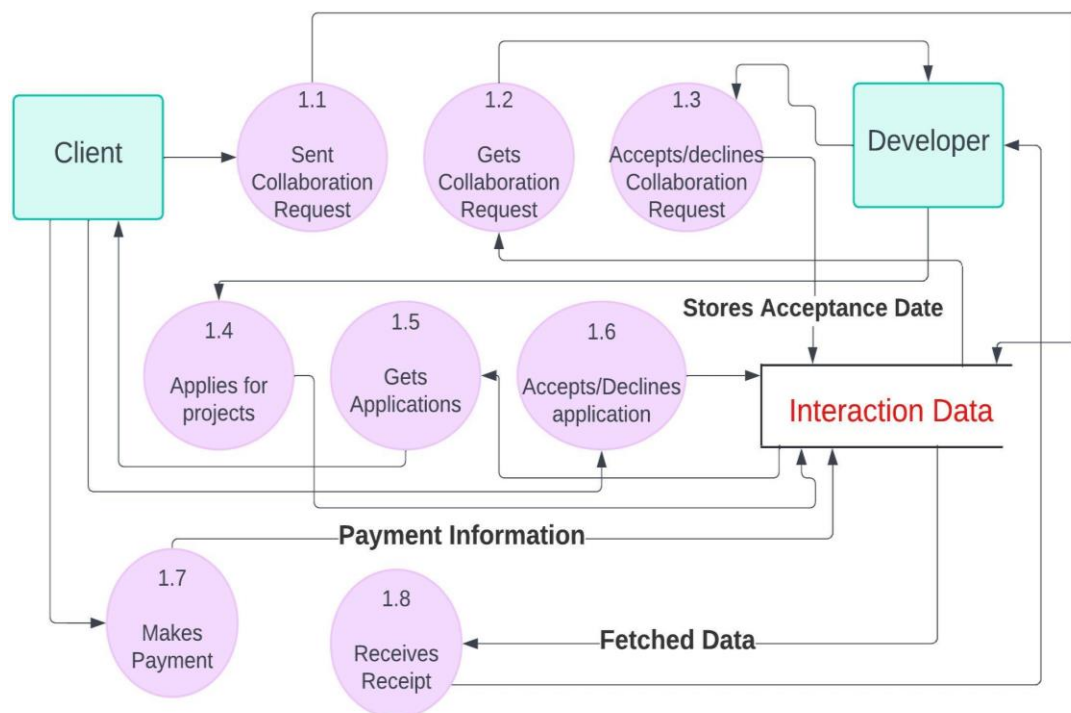
Process:

- 1.1. Post projects
- 1.2. Edit projects
- 1.3. Browse projects
- 1.4. Search projects
- 1.5. Filter projects

Activities:

- ✓ Clients can post projects that will be stored in database immediately
- ✓ Clients can edit them if they want to which will update existing project data in database
- ✓ Potential Developers can behave like clients and post projects
- ✓ Developers typically has the facilities to browse, search and filter out project data fetched from database according to their skills and interest

3. Interaction management:



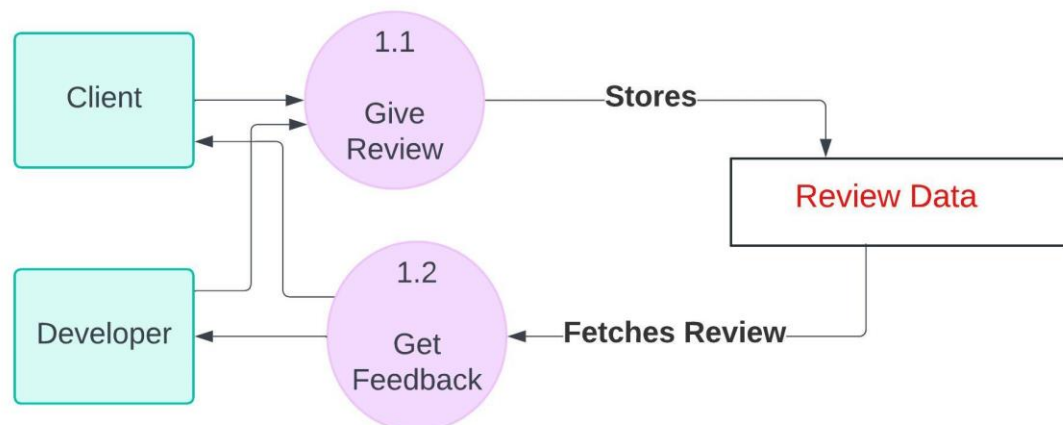
Entities: Clients, Developers

Process:

- 1.1. Sent collaboration request
- 1.2. Gets collaboration request
- 1.3. Accepts/Declines collaboration request
- 1.4. Applies for projects
- 1.5. Gets Applications
- 1.6. Accepts/Declines application
- 1.7. Makes payment
- 1.8. Receives Receipt

Activities:

- ✓ Clients can send collaboration request to their desired developers via our special “Handshake” feature.
- ✓ Developers will get notified and they can accept or decline the requests which will notify clients about the dealing. If accepted database will store the date as the starting date of the project deal to display it on the “Work History” section of the platform of that certain developer.
- ✓ Developers can apply for project tasks
- ✓ Clients will get notified about the applications and they can accept or decline which will notify developers thereby. Again if accepted database will store the date as the starting date of the project deal to display it on the “Work History” section of the platform of that selected developer.
- ✓ Once the assign deal finishes clients will pay all the developers who worked via payment gateway and database will store the payment information including date declaring it as the deal ending date. Later this information will be used to provide the developers with payment receipts.

4. Review management:

Entities: Clients, Developers

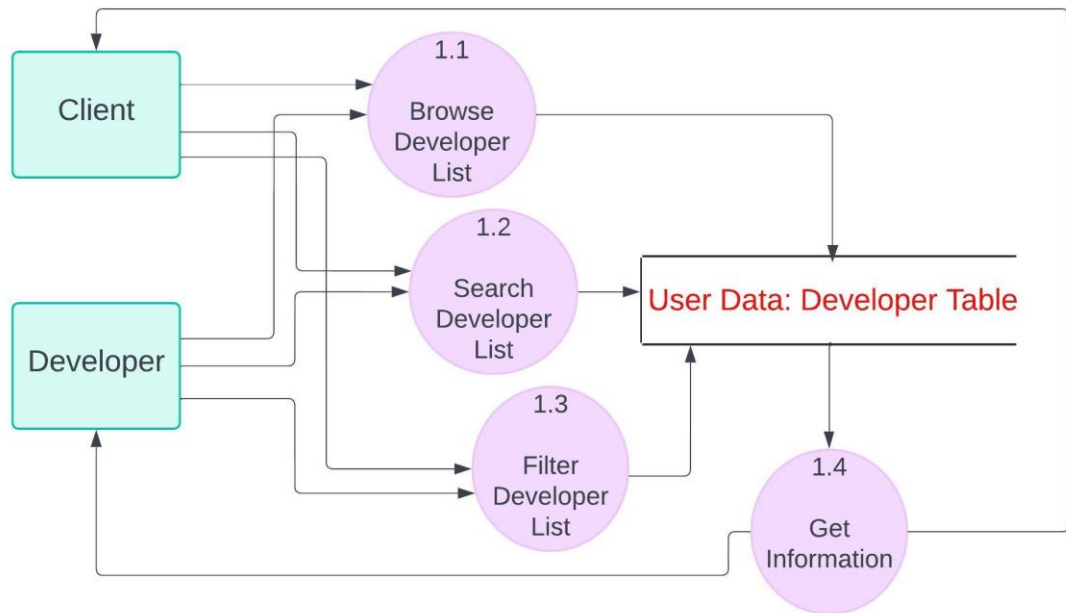
Process:

- 1.1. Give Reviews
- 1.2. Get Feedback

Activities:

- ✓ Both clients and developers can give and get feedbacks on their experience of working with each other.
- ✓ This feedbacks will be stored in the Review data section of the database and later fetched to display it on the platform.

5. Exploration management:



Entities: Clients, Developers

Process:

- 1.1. Browse Developer List
- 1.2. Search Developer List
- 1.3. Filter Developer list
- 1.4. Get Information

Activities:

- ✓ Both clients and developers can browse developers list as clients need developers and developers might need to team up for certain projects respectively.
- ✓ Again both of them can search and filter developer accounts from the list according to their preferences.
- ✓ This browsing, searching, filtering processes will access the database to take out all the information from user data section where the accounts belong to developers.

Conclusion:

The Data Flow Diagram for AllyHub gives a precise visualization on how data flows within the system between its processes and entities. It also breaks down the complex processes into small understandable parts, directing how clients and developers interact within the system. Therefore, the Data Flow Diagram maps the flow of data in the system more precisely and clearly.