Architecture

- Do not begin the decomposition of a system into modules on the basis of a flow chart
- Instead, begin with a list of difficult design decisions of design decisions that are likely to change
- Each module is then designed to hide such a decision from the others
- Modules will often not correspond to steps in the processing
- To achieve an efficient implementation we must abandon the assumption that a module is one or more subroutines, and instead allow subroutines and programs to be assembled collections of code from various modules

Difficult design decisions:

1. Username lookup in database
2. Friend List lookup in database
3. Suggested friend list compilation
4. Group Userlist lookup in database
5. Providing content such as images and how to provide them.
6. Storage of profile and group content
    a. Dont know how to keep track of users and groups

Design decisions that are likely to change

1. Various database lookup styles
2. Relational queries based on JSON styled Database, MongoDB.
3. API calls within the database to acquire data.

Entity Management (Devin)

Description:

(Client-side JavaScript) Account creation, management, authentication, and modification associated with guests, registered users, group admins, and group members.

Inputs

- Username
- Password
- Email address
- Generic personal information
- Responses from user database

Processing

- Database requests to create account based on supplied information
- Database requests to modify account based on supplied information
- Database requests to create group admin based on supplied information
- Database requests to modify group admin based on supplied information

Outputs

- Update status

Module-Module Interfaces

- User Database
- User interface

Entity Database (Devin)

Description:

(MongoDB, Node.js) Storage, updating, retrieval, and back-to-front transfer of user data which includes lists of registered users and their associated profiles, lists of group admins and their associated groups, lists of members belonging to groups, lists of friends belonging to a user, and the personal data associated with each user

Inputs

- Request for friends list
- Request for user profile information
- Request for user account information
- Request for group information
- Request for group administrator information
- Request for group member information
- Request to form friendship
- Request to modify user profile information
- Request to modify user account information
- Request to modify group information
- Request to modify group administrator information
- Request to modify group member information
- Request to create account
- Request to create group

Processing

Outputs

1. User profile/user account/group/group administrator/group member/friend information
2. Modification success status
3. New user account created
    a. Verification e-mail
4. New group

Module-Module Interfaces

- User Access Management

User Interface (Kyle)

Description:

HTML and CSS files that comprise the collection of user-interactable input sources (buttons, text fields, hyperlinks)

Inputs

- Text

- Mouse clicks

Processing

- Modify interface according to client-side scripts following user interaction

Outputs

- Modified user interface

Module-Module Interfaces

- User Access
- Content Access

Page Server (John)

Description:

Respond to requests made from front end for HTML pages and their associated content

Inputs

- Clicks on hyperlinks

Processing

- HTML page and CSS referenced by click is retrieved
- Backend content supplied where necessary

Outputs

- Client browser is updated with retrieved HTML and CSS

Module-Module Interfaces

User Interface

Content Retrieval (Karl)

Description:

Handle client-side requests for new content such as pictures, polls, posts, comments, suggest friends lists by modifying the user interface and using data retrieved from the user database and content server

Inputs

- Text
- Clicks

Processing

- Run scripts initiated by user actions

Outputs

- Modify user interface with or without data retrieved from user database and content server

Module-Module Interfaces

- User interface
- Content Server
- User Database

Content Server (John)

Description:

Store, maintain, modify, and update photos, posts, comments, polls

Inputs

- Request for content
- Request to create/modify/update content

Processing

- Update the content
- Upload or save modified content

Outputs

- Requested content

Module-Module Interfaces

- Content Management
- Content Modification

Content Modification (Karl)

Description:

Modify content upon requests from the user interface

Inputs

- Request to modify content
- Content modification

Processing

- Modify content in content server

Outputs

- Content server modified
- User interface updated

Module-Module Interfaces

- User interface
- Content server

Suggested Friends List Curator (Devin)

Description:

Detect active user and determine list of suggested friends from user database based on several criteria

Inputs

- Main portal open

Processing

- User’s friend list is searched

Outputs

- Most common friends of friends is listed

Module-Module Interfaces

- User Interface
- User Database