# Team Dec 14-13
# CySwap
# Project Plan

Client: Merry Rankin
Adviser: Professor Mitra

Team Members:
Fabian Briesmoore
Jared Cook
Kyle Johnson
Adam Sunderman

# Table of Contents

# Problem/need Statement

The Government of the Student Body is aiming to develop a website that allows Iowa State Students to buy and sell goods. The site will allow for multiple types of items to be sold including: textbooks, furniture, clothes, tickets, places for rent, etc. Each type of post will need to be organized in ways that make sense for the type of item, for example, books will be searchable with ISBN number and the website should fill in all relevant information accordingly. The site will not handle the monetary transaction between students. It should instead be a way of connecting students. However, it should use a formalized procedure for contacting and completing transactions to allow for detailed reporting features. To build a strong sense of community, the web-site should be accessible through the use of netid (@iastate.edu emails).

# System Block Diagram
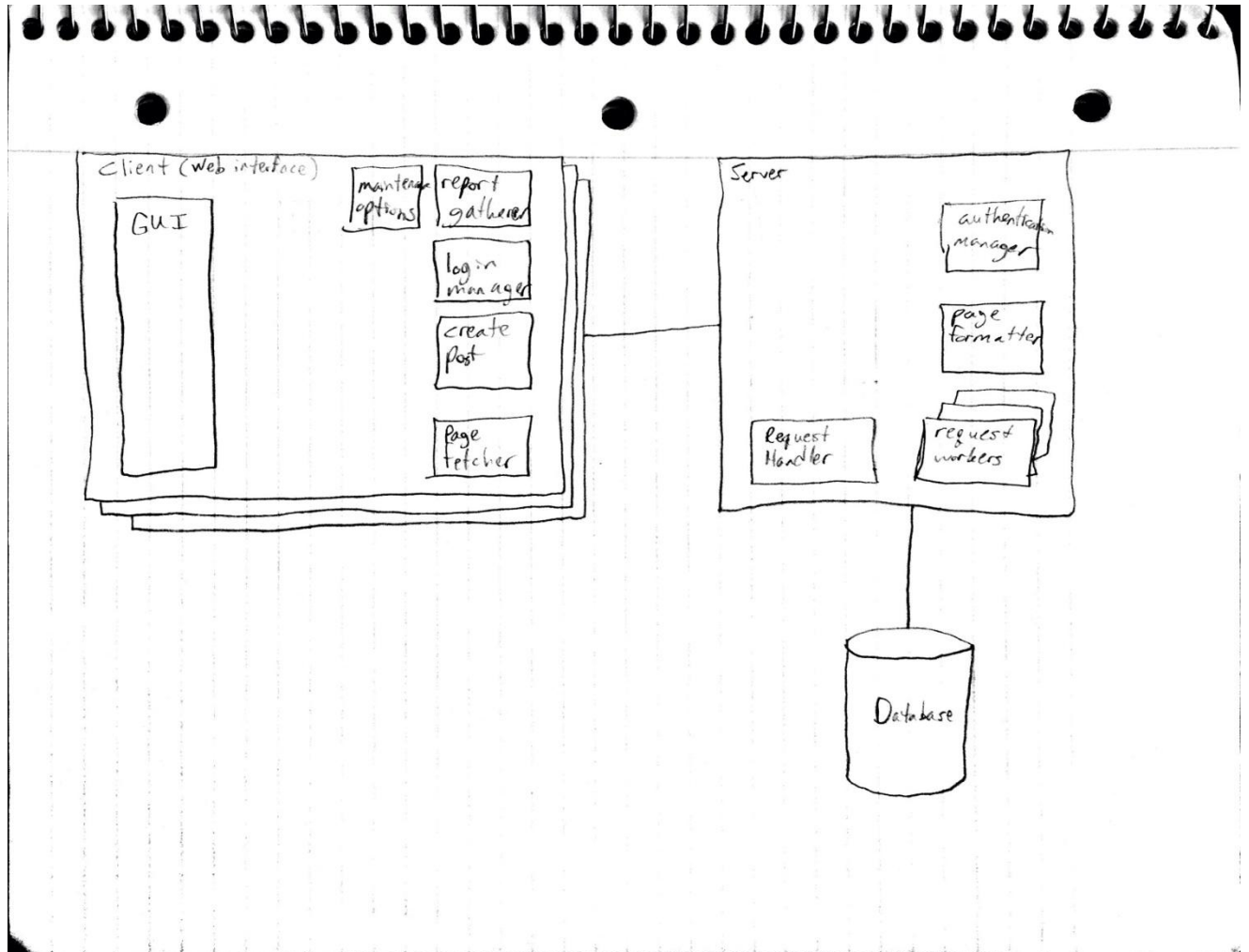
## System Description

We aim to develop a web based application to allow users to view and post listings for several different types of items for sale with fellow Iowa State students. Posts will be stored in the server's database for future access from any potential buyers. The users will be able to find posts relevant to them through search categories (books, tickets, furniture, etc.) as well as search filters (price, popularity, etc.).

The transactions will be handled independently from our system. However, the system will implement a formalized process for making contact with a seller to ensure accurate reporting statistics within our system as well as help automate post management (removing completed transactions and keeping track of item interest and popularity). Upon viewing a post, a user will be able to contact the seller via a "make an offer" button. They will then fill in a form that will allow the system to complete an auto-generated email to the seller with the offer. The system will keep track of this interaction and others like it to gauge interest and popularity of an item. If the post has not been closed a week after contact, our system will auto-generate and send an email to inquire if the transaction had been completed, and update internal information accordingly. Once a transaction is complete, automatically generated emails will ask the buyer and seller to submit feedback on each other. The system will keep track of the successful transaction in the database for reporting purposes.

When users post an item, the system will allow them to choose a category. Upon selecting a category the system will use an appropriate form to collect the necessary post information as well as help automate the process for items with standardized data ,for example, ISBN numbers for books will help auto-complete the majority of the form for the user. Upon the form completion the item will have a posting generated and stored within our database. The post will be accessible for any users searching for the item.

In addition, The Government of the Student Body will be able to use the system with additional functionality including: gathering reports on the usage of the system (general traffic, interest levels per category, successful transactions, average feedback per transaction, etc.) as well as adjusting high-level options to govern the system's operating procedures for high-level maintenance with minimal interaction at a coding level.

## Operating Environment

Our website will be mobile friendly and should be accessible by phones running Android or iOS. We will have a database that will store information about users and store other necessary information. We also may be accessing the Iowa State Bookstore and Iowa State Athletics databases.

# User Interface Description

## Template Pieces

*Header*

- The header will consist of two rows. The top row will contain the logo, site name, search bar, and log in button.
- The bottom row will be the navigation bar. The main pages to navigate to here will be Home, Categories, Safety, and About Us

*Footer*

- The footer will contain links to Copyright information, terms of use, reporting content, and contacting the website managers.

# Pages

*Home Screen*

- The home screen will contain a select few postings from the most popular categories.

## Category Search Screen

- The Category Search Screen will show all posts from the selected category. It will have filtering options in the right column, and have the subsequent pages at the bottom.
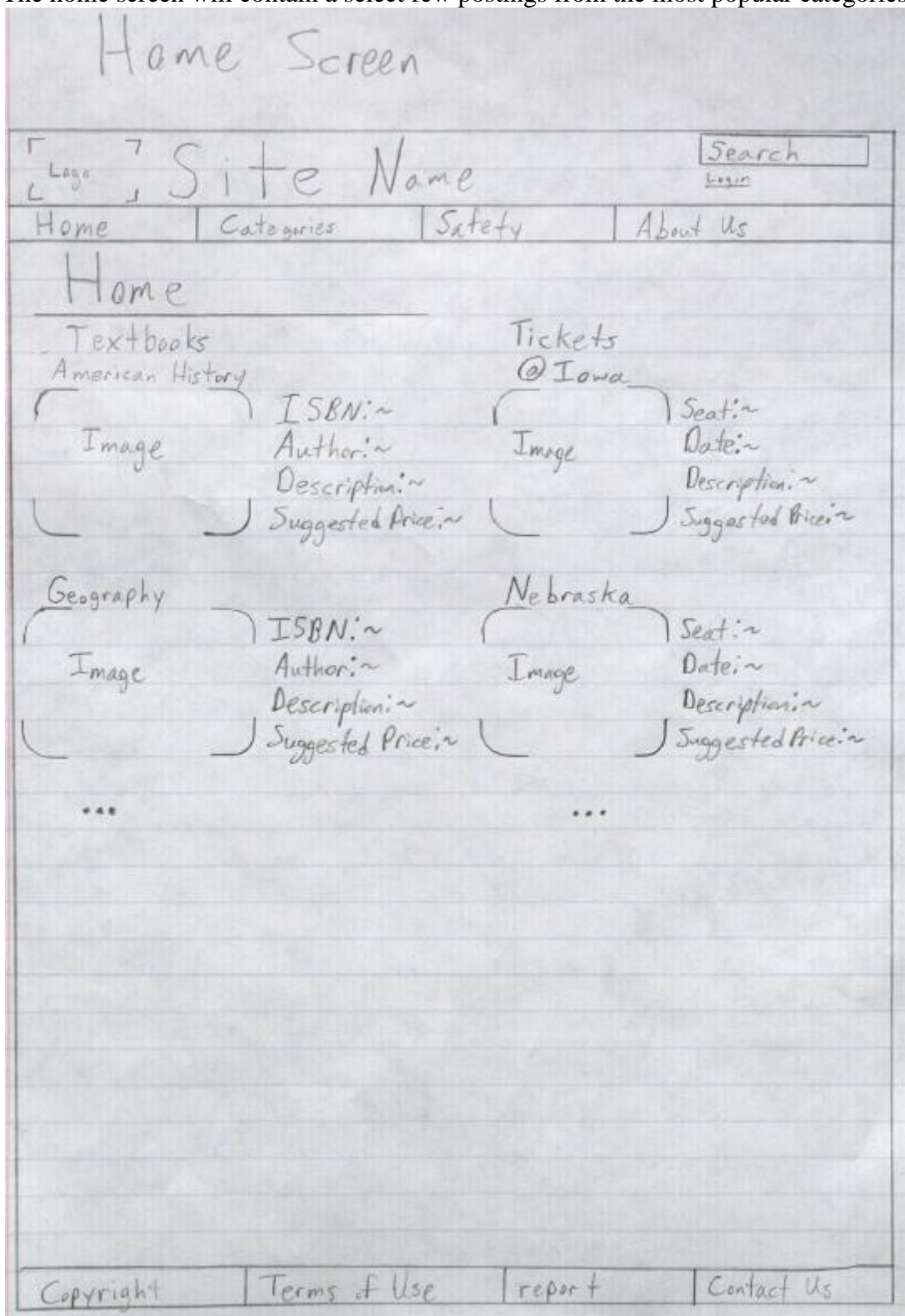
*Post Screen*
- The screen for an individual post will contain pictures of the item, suggested price, information about the item, a description, the poster's previous feedback, and a Contact Poster button.
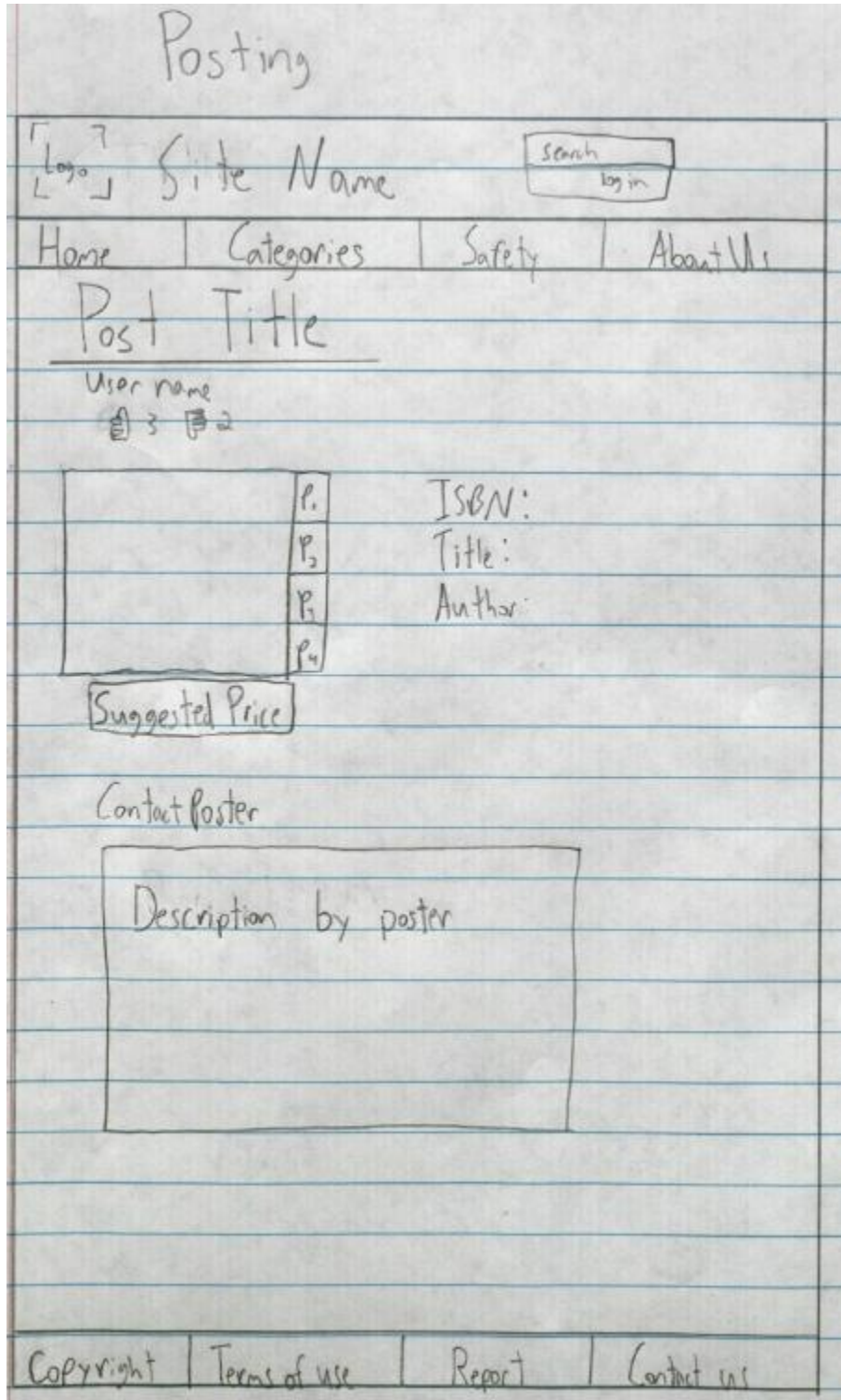
# Functional Requirements

- Interface
    - Code must be easily understood and maintained
    - Must be visually appealing
    - Allow users to flag inappropriate content
    - Includes a posting management system that allows users to edit their posts
    - Posts will only be visible to logged in users
    - Automatically removes posts over 60 days old unless relisted
    - Removes postings that have been sold, possibly via email confirmation
    - Has a section for tips, including appropriate locations and police contact information
    - Allows students to browse/sort/filter by price for all postings
    - Textbooks
        - Able to browse textbooks by department, class, and section
        - Can sort by condition
    - Tickets
        - Can sort by sport type and game
        - Can sort by ticket type (e.g. online transfer, print, etc.)
- Admins
    - Admins will have the ability to edit any posted content
    - An email should be sent to an admin if content is flagged
- Backend
    - Able to handle web traffic during high trading periods, such as the beginning and end of the semester
    - Easily able to update ISU curriculum and textbook lists (excel list)
    - May need to work with ISU administration / provost to get copy of list
    - Only allow users with "@iastate.edu" email addresses
    - Account username and password will be the same as a student's net-id and password

# Non-functional Requirements

- Must be visually appealing
- Needs to adapt to Iowa State's website requirements (header, footer, color scheme)

# Resource Requirements

- **Hardware**
  - Server
- **System Software**
  - SQL
  - PHP
  - Java
  - JavaScript
  - CSS
  - HTML
- **Development Software**
  - Atlassian Software

# Work Plan

## Deliverables

| Deliverable | Date |
|---|---|
| Design Document Version 1 | 03/14/14 |
| Project Plan Version 2 | 04/04/14 |
| Design Document Final Version | 04/25/14 |
| Project Plan Final Version | 04/25/14 |
| Presentation | 04/29/14 |

## Risks

There could be issues getting access to databases for our project. Additionally, if we get access and our software relies on databases having data formatted in a certain way, there could be issues when this format is changed.

It is possible that a team member will leave our project. This risk will require us to maintain thorough documentation through the development of this project.

We will have to be prepared to handle large amount of traffic. We will likely need to be able to process significant traffic during the start of each semester, when students are buying textbooks.

## Team Member Roles

| Team Member | Role |
|---|---|
| Fabian Briesmoore | Key Idea Holder |
| Jared Cook | Webmaster |
| Kyle Johnson | Team Leader |
| Adam Sunderman | Communicator |

## Project Schedule

| Task Name | Start Date | End Date | Duration | Predecessors |
|---|---|---|---|---|
| **Course Deliverables** | **03/06/14** | **04/29/14** | **39** | |
| Design Document Version 1 | 03/06/14 | 03/14/14 | 7 | |
| Project Plan Version 2 | 03/27/14 | 04/04/14 | 7 | |
| Design Document Final Version | 04/14/14 | 04/25/14 | 10 | |
| Project Plan Final Version | 04/14/14 | 04/25/14 | 10 | |
| Presentation | 04/14/14 | 04/29/14 | 12 | |
| **Product Development** | **03/06/14** | **12/12/14** | **202** | |
| Create Iteration Plan | 03/06/14 | 03/19/14 | 10 | |
| Version 1 | 03/20/14 | 04/25/14 | 27 | |
| Write User Stories | 03/20/14 | 03/26/14 | 5 | 8 |
| Develop Code | 03/27/14 | 04/17/14 | 16 | 10 |
| Test Code | 04/18/14 | 04/24/14 | 5 | 11 |
| Version 1 Release | 04/25/14 | 04/25/14 | 1 | 12 |
| Version 2 | 08/25/14 | 10/17/14 | 40 | |
| Write User Stories | 08/25/14 | 09/04/14 | 9 | |
| Develop Code | 09/05/14 | 10/02/14 | 20 | 15 |
| Test Code | 10/03/14 | 10/16/14 | 10 | 16 |
| Version 2 Release | 10/17/14 | 10/17/14 | 1 | 17 |
| Version 3 | 10/20/14 | 12/12/14 | 40 | |
| Write User Stories | 10/20/14 | 10/30/14 | 9 | 18 |
| Develop Code | 10/31/14 | 11/27/14 | 20 | 20 |
| Test Code | 11/28/14 | 12/11/14 | 10 | 21 |
| Version 3 Release | 12/12/14 | 12/12/14 | 1 | 22 |

## Project Schedule Gantt Chart

A Gantt Chart of the Project Schedule in .png form is available here.