

## **Project Abstract**

The BSC (Berkeley Student Cooperative) houses are based around a communal sharing of labor, administered by one or two workshift managers (WSMs) at each house. The residents and WSMs require a software system to manage the complicated process of allocating and assigning shifts, marking shifts as complete, and fining residents who do not complete their shifts.

Each resident of the house uses the Workshift Manager's Assistant (WSMA) to submit a form specifying which shifts they are available to do and which they prefer. The WSM then assigns chores to each resident, using passive advice (or active calculation) offered by the system, such that the scheduling constraints, and as much as possible the preferences, are met. For the rest of the semester, the WSM and the system track the progress of the chores. After completing a shift, a resident signs off for the task. The system fines residents who get too far behind on their work hours according to a fining schedule and policies set by the WSM.

## **Document Revision History**

Revision 1.1 2011-10-26

## **Customer**

The customer of the WSMA is the Central Office of the BSC. The users are the residents and WSMs of the individual houses. The system is being tailored specifically to the needs of the BSC, but other coops with similar workshift paradigms may find it useful as well. The specific users we are working with to develop the system are Natalie Briggs, WSM at Afro House and Ariane Lange, WSM at Stebbins.

## **Competitive Landscape**

The BSC already has a WSMA that was created by Janak Ramakrishnan at UC Berkeley seven years ago. The system, while it does provide the bulk of the functionality that the BSC requires, has severe usability problems. Users of the current system, both residents and workshift managers, are eager for a new and better version. Since our WSMA will be the only workshift management tool designed specifically for the BSC, with all the custom features the coops need, it has no direct competitors in that respect. However, there are software systems out there that provide employee time management services of one sort or another.

One such system is ShiftPlanning, a combination time-clocking/scheduling/payroll system for businesses. It provides much of the functionality that the BSC needs, but it also has many

extraneous functions that the BSC would be paying quite a lot of money not to use. ShiftPlanning would cost the BSC hundreds of dollars per month, putting extra financial burden on house residents and defeating the purpose of coop living: cheap lodging for students in return for their work helping to maintain the residence. ShiftBoard and SnapSchedule are other shift schedule management systems that are very similar to ShiftPlanning, but without the payroll and other extra functions, focusing on just scheduling and messaging. This is closer to what the BSC needs, but again they both cost a considerable amount of money which would translate into additional fees for residents.

Our system, although it has the disadvantage of taking more time to create than it would take to buy a pre-made one, is the best solution for the BSC because we will do it at no cost to them and because it will be tailor-made for them, with all the specific features, terminology, and optimizations for scheduling coop workshifts.

## User Requirements

There will be two kinds of users that will use our system. WSMs use the system to initially assign shifts to residents and set system parameters, such as chore details and fining periods. Afterwards, WSMs monitor the progress of weekly shifts. If using online sign offs (optional) the system will keep track of completed shifts automatically. If not, the WSM must manually enter which shifts have been completed and which haven't. WSMs are able to create special shifts that are open to all residents to sign up for.

The other users are the residents of the house. They will need to be able to login to their account, from which they submit their preferences for shifts, see what shifts they have been allocated, see how many hours they are up or down, sign up for available shifts, sign out from shifts which they are unable to complete, sign off that they completed their shifts, and verify that somebody else has done workshift (given username/password).

### WSM account

1. if there exist no WSM accounts, option to create one
2. login
  - a. prompt for username and password
  - b. link to administration panel
  - c. display link to logoff
3. login using JomSocial account (extra feature)
4. logoff
5. change your password
6. change system parameters
  - a. start a new semester (clears the database)
  - b. set up the dates of the semester
  - c. set the number of hours each resident is required to complete per week

- d. set the number of hours each resident is required to complete in a specific week
  - e. set up preference form options
  - f. upload a workshift policy document
  - g. turn on or off online sign ups
  - h. set up online sign off settings
    - i. require verification from another user
    - ii. how long the resident has after the chore deadline to sign off online
  - i. set up blow-off options
- 7. manage house roster
  - a. view directory with all information available (regardless of privacy settings)
  - b. import a list of residents from CO
    - i. system will create an account for each with a random password
    - ii. system will email the default password to each resident
  - c. add new resident
  - d. delete resident
  - e. rename resident
  - f. view and edit profile details for a particular resident
    - i. preference form response
    - ii. contact information
    - iii. room number
    - iv. number of hours the resident is required to complete per week
    - v. number of hours the resident is required to complete in a particular week
  - g. reset a resident's password
  - h. make a resident's account into a WSM account
  - i. view a resident's shift history for the semester
- 8. configure chores
  - a. create or edit chore: input name, description, number of hours, how long in advance a sign out is required
  - b. delete chore
- 9. configure shifts
  - a. create or edit shifts (instances of chores): input date and time (if relevant)
  - b. delete shifts
- 10. allocate shifts
  - a. manually allocate shifts
    - i. choose how much help the interface gives you when allocating shifts

- ii. see a list of all residents, amount of hours allocated per week to each, and amount of hours each is obligated to complete per week
  - iii. display passive advice based on resident preferences, amount of hours allocated to each resident, etc
  - iv. allocate shifts to residents
- b. automatically allocate shifts (extra feature)
  - i. automatically allocate shifts to residents, fulfilling all constraints and maximizing satisfaction of resident preferences
- c. print a sign off sheet if not using online sign offs

#### 11. manage shifts

- a. calendar view of work shift allocations
- b. calendar view of work shift assignments for a given week
  - i. see shift status (pending, completed, available, blown off)
  - ii. add a custom shift
  - iii. edit a shift
- c. view shifts allocated to a specific resident
- d. view shifts assigned to a specific resident
- e. view shifts which are not assigned to anyone
- f. modify shift assignment
  - i. mark shift as completed, pending, or blown off
  - ii. assign an available shift to a resident
  - iii. unassign a shift from a resident

#### 12. fines

- a. set fining schedule
- b. adjust fining settings
- c. see a list of all residents' fines (date, reason, amount) and their status (paid or unpaid)
- d. see a list of a specific resident's fines and their status
- e. manually fine a resident
- f. manually remove fine from a resident
- g. mark a fine as paid
- h. export a record of all fines for the semester

#### 13. backup (used to archive and view data from past semesters, and for safety) (extra feature)

- a. create a backup archive of the database
- b. restore database to the state of a backup
- c. delete a backup

### **Resident account**

#### 1. login

- a. prompt for username and password

- b. link to administration panel
  - c. display link to logoff
- 2. logoff
- 3. profile details
  - a. contact information (view and edit)
  - b. room number (view and edit)
  - c. amount of hours the resident is obligated to complete per week (view)
  - d. change password
  - e. change privacy details to control which details will show up in the directory
- 4. change preferences
  - a. rank each shift from 0 (cannot do) to 5 (highly prefer)
- 5. view work shift allocation
- 6. manage assigned shifts
  - a. see a list of assigned shifts
  - b. see shift statuses (pending, completed, available, blown off)
  - c. online sign off (marks it as completed)
    - i. get another user to authenticate to verify your completion (optional)
  - d. sign out of a shift (unassign it from oneself)
  - e. see how many hours you are up or down
- 7. see available shifts
  - a. sign up for an available shift (assign it to oneself)
- 8. see shift history
  - a. view all shifts assigned in the past, and their status
  - b. view the total amount of hours required to complete, assigned, completed, and blown off
  - c. see shifts the resident has verified that another resident has completed
- 9. fines
  - a. see a list of fines (date, reason, amount) and their status (paid or unpaid)
  - b. see the fining schedule and policy
- 10. see the directory of residents and how many hours each resident is up or down
- 11. see the BSC announcement twitter feed

## Use Cases

Name	View roster (must have)
Actors	Workshift manager
Triggers	Click on “view roster” button
Events	Brings workshift manager to a page where they can select a resident profile to view
Exit condition	Navigate to another page
Post-conditions	Workshift manager should be able to view resident profiles, add or delete residents
Acceptance test	A workshift manager would navigate to this page from the home page. Once here, they should be able to select a resident profile from either a list or a drop down menu.

Name	Create chore (must have)
Actors	Workshift manager
Triggers	Click on “create chore” button
Events	Opens a form and allows the workshift manager to input name, description, number of hours, and how long in advance a sign out is required
Exit Condition	Click “ok”, chore is created
Post-Conditions	Chore is listed on a master list of chores
Acceptance test	A created chore should be listed on the master list of chores and shifts should be able to be made using said chore

Name	Create shift (must have)
Actors	Workshift Manager
Triggers	Click on “create shift” button
Events	Opens a form and allows WSM to input date and time of a previously created chore
Exit Condition	Click “ok”, shift is created
Post-Conditions	Shift is displayed on the calendar view of shifts

Acceptance test	A created shift should be displayed on the calendar view of shifts and it should be assignable to a resident
-----------------	--

Name	Allocate shift (must have)
Actors	Workshift manager
Triggers	Click on “allocate shifts” button in ‘manage shifts’ page
Events	Arrive on allocate shift page from the manage shifts page and drag a resident name to a workshift slot
Exit Condition	Drag a student to a workshift slot, navigate to a different page
Post-Conditions	Resident name now fills the work shift slot
Acceptance test	Navigate to allocate shift page. Drag a resident name to an unallocated shift. Resident should now be in the workshift time slot.

Name	Sign out of shift (must have)
Actors	Resident
Triggers	Click on shift, choose “sign out” button
Events	Pop up window appears and allows resident to sign out of a shift they are assigned to
Exit Condition	Click “ok”, shift is removed from assigned shifts
Post-Conditions	Shift is removed from resident’s assigned shifts, shift status is changed to unassigned in WSM’s view
Acceptance test	The removed shift shouldn’t be visible on the resident’s shifts page and the shift’s status should be unassigned

Name	Sign off a workshift (must have)
Actors	Resident
Triggers	Navigate to workshift page
Events	Navigate to workshift page, submit number of hours completed, have verifier (another resident) input their name and password, click sign off button
Exit Condition	Click “sign off”, navigate to different page

Post-Conditions	Shift is marked as signed off
Acceptance test	Navigate to workshift page. Click on workshift. Input number of hours completed. Have verifier input name and password. submit the form.

Name	Import list of residents (must have)
Actors	WSM
Triggers	Click on “import roster” button
Events	Opens a form and allows WSM to choose a file to upload System will add all residents on list to internal database
Exit Condition	Click “ok”, roster should be updated
Post-Conditions	Roster contains contents of uploaded list
Acceptance test	A list of residents can be uploaded and then you can check the roster to see if all the residents were added

Name	Set the parameters (must have)
Actors	Workshift Manager/Residents
Triggers	Click on “edit” button
Events	The current page becomes editable
Exit Condition	Click “ok”, the information is updated
Post-Conditions	The information gets updated
Acceptance test	Check if the information is changed after one modification

Name	Residents fill in/update the workshift preference form (must have)
Actors	Residents
Triggers	Click on the “Workshift preferences” button on the home page
Events	Resident is taken to the “Workshift preferences” page, can see list of all workshifts and can rate each shift as one of: “Absolutely cannot or will not do”, “Don’t want to do”, “Neutral”, “Would like to do”, “Really want to do”, can also see previous ratings that they have given to shifts
Exit Condition	Resident clicks the “Save preferences” button
Post-Conditions	Resident is returned to the home page and their preferences are



	recorded in the system
Acceptance test	Resident can go to the “Workshift preferences” page, see any previous ratings they gave for shifts, set their preference for any shift, and the workshift manager can see the resident’s preferences as well

Name	Add people to roster (must have)
Actors	Workshift Manager
Triggers	Go to manage roster page, click on “add resident” button
Events	Text box allows WSM to add one or more residents to roster
Exit Condition	Click “ok”, residents are added
Post-Conditions	Roster includes new additions
Acceptance test	Sample residents can be added, see if roster reflects changes

Name	Change the default password (must have)
Actors	Residents
Triggers	Click on “change password” button on the profile page
Events	Users enter a new password/confirm the password
Exit Condition	Click “save”, new password is stored
Post-Conditions	The password of the user gets changed in the system
Acceptance test	Log in the user’s account using the new password and check if the login is successful

Name	Backup system (extra)
Actors	Workshift Manager
Triggers	Click on “System Backups” in the “Administrative tools” page
Events	WSM clicks on “System Backups”, gets taken to the “System Backups” page, clicks on “Create new backup”, sees a message saying “Backup in progress...”, once backing up is done message disappears
Exit Condition	The backup process finishes
Post-Conditions	A backup of the system state at the current time is created and accessible from the list of system backups

Acceptance test	WSM can create a backup and see it in the list of all backups
-----------------	---

Name	Change the number of hours a resident owes (must have)
Actors	Workshift Manager
Triggers	Click on “edit” button on the particular resident’s page on the house roster
Events	WSM goes into the house roster, finds the resident and views the detail of the resident.
Exit Condition	Click “submit” button at the end of the page.
Post-Conditions	The hours of the particular residents gets changed to whatever the WSM modifies to.
Acceptance test	Change the weekly hour of this particular resident through the admin account. The particular resident logs into his/her homepage and checks if the weekly hour is changed.

Name	Set the fining period (must have)
Actors	Workshift Manager
Triggers	Click on drop-down “Set the fining period” menu in “Fines” page
Events	WSM clicks on the “Fines” link from the home page, gets taken to the “Fines” page, clicks on the drop-down menu with the title “Set the fining period and selects one of “Every week”, “Biweekly”, “Every month”, “Three times a semester”, “Twice a semester”, and “At the end of the semester”
Exit Condition	WSM selects an option from the drop-down fining period list
Post-Conditions	The fining period is changed to what the WSM selected
Acceptance test	The WSM can see the previous setting for the fining period and set a new one, fines are tallied at the end of each fining period, and residents are notified of approaching fining period deadlines

Name	View the history of one’s total workshifts done (must have)
Actors	Resident
Triggers	click on the link that says “list of all shifts done”
Events	Opens a new page showing tables of all workshifts finished since week

	1
Exit Condition	Click "ok"
Post-Conditions	N/A
Acceptance test	Sign off a workshift and check if the change of the total workshift reflects on the history

Name	Set the default fine amount (must have)
Actors	Workshift Manager
Triggers	Click on "Edit" button next to the default fine amount listed on the "Fines" page
Events	WSM clicks on the "Fines" link from the home page, clicks on the "Edit" button next to the default fine amount listing, the listed number turns into a text box with the number in it, WSM deletes the number and types in a new one, clicks the "Set" button
Exit Condition	Click on the "Set" button
Post-Conditions	The default fine amount is set to the value the WSM specified and all residents will be fined that amount for each hour they are down
Acceptance test	The WSM sets the default fine amount, can see the correct amount on the "Fines" page, and resident's fines are calculated using that amount

Name	Manually fine a resident (must have)
Actors	Workshift manager
Triggers	Click on "Manually fine a resident" in the "Fines" page
Events	WSM navigates to the "Fines" page from the home page, clicks on "Manually fine a resident", gets taken to a page where they select the resident's name from a drop-down menu, enter the fine amount in a textbox, and enter any notes about the reasons for the fine in another textbox, finally clicks on the "Fine" button
Exit Condition	Click on the "Fine" button
Post-Conditions	The resident in question has a fine in the specified amount applied to their account
Acceptance test	WSM can set a fine for a resident and then the WSM and the resident can both see that reflected in the resident's profile

Name	Create special shift (must have)
Actors	Workshift manager
Triggers	Click on “Create special shift” in the “Manage shifts” page
Events	WSM navigates to the “Manage shifts” page from the home page, clicks on “Create special shift”, gets taken to a page where they enter the shift name, description, and (if applicable) time slot into textboxes, selects from a calendar view how long the shift will exist, clicks on “Create shift” button
Exit Condition	Click on “Create shift” button
Post-Conditions	A new shift is created with the title, description, time slot, and date span chosen by the WSM
Acceptance test	WSM creates a special shift and the WSM and residents can see it in the list of all shifts

Name	Manually sign off a resident (must have)
Actors	Workshift manager
Triggers	Click on “Manage shifts” link from the home page
Events	WSM navigates to the “Manage shifts” page from the home page and selects “completed” from the drop-down menu next to the listing of the shift and resident’s name
Exit Condition	Selecting the “completed” option from the drop-down menu
Post-Conditions	The resident will be marked as having completed the shift
Acceptance test	WSM can set a resident’s shift to “completed” and the resident does not see any down hours recorded from that shift on their account

Name	Resident change their password <b>(added in version 1.1)</b>
Actors	Workshift manager and General users
Triggers	Click on “edit” link
Events	A user checks their profile page. There is a text box for them to input a new password, and another form for them to confirm the password.

Exit Condition	Clicking on the “new password” button.
Post-Conditions	The users’ password will be updated in the database.
Acceptance test	A user changes their password. They log out, then log back in with the new password. If they can successfully log in, then the test passes.

## User Interface Requirements

The user interface is the primary challenge of this project. The assignment and management of work shifts is a complicated process, but our interface must make it easy for the user to visualize. The interface will shape both the user’s workflow and the user’s mental model of how the workshift system works.

The users of WSMA will be mainly the students (in average, aged from 18 to 23) who attend schools in or near the city of Berkeley. There are two user groups for the system: one is the residents who play the role of users of the system; one is the WSMs who play the role of the administrators. Therefore, the user interface must be easy to learn/manipulate and straightforward, similar to an online schedule planning tool (such as course ninja), but with extra features.

To integrate an explanation of what each feature does directly into the interface, we will have question mark icons next to many UI elements. They are unobtrusive enough that they do not get in the way of normal usage, but mousing over them will reveal a explanation of the feature, what it does, and how it is used.

## House Roster and Resident Profiles

Berkeley Student Cooperative Workshift Manager

[Home](#) [Profile](#) [Logout](#)

**STUDENT**  
**8op**  
**BERKELEY**

**Workshift**  
**Manager**

[My Shifts](#)  
[Shift Preferences](#)  
[My Fines](#)  
[Allocate Shifts](#)  
[Monitor Shifts](#)  
[Chores](#)  
[Manage Fines](#)  
[House Roster](#)  
[Administrative Tools](#)

### Profile

Name	Charles Bingley	<a href="#">Change</a>
Room Number	18	<a href="#">Change</a>
Email	bingley@netherfield.com	<a href="#">Change</a>
Phone Number	(212) 345-1813	<a href="#">Change</a>
Password	*****	<a href="#">Change</a>

### Profile View mockup (added in version 1.1)

When the resident logs into the profile, they will be able to see the basic information (email address, password, phone number, etc.) and edit them. On the page of the profile, there are links to the workshift preference form where the users are able to view/edit them, to the history of all workshift done/verified in the semester, to the workshift allocation page where users can sign off/out of a workshift, to the shifts management page where the users can take the available shifts that are either created specially or signed out by someone else and to the penalty page where the users can check the net hours and the corresponding amount of fines.



## Workshift Manager

[My Shifts](#)[Shift Preferences](#)[My Fines](#)[Allocate Shifts](#)[Monitor Shifts](#)[Chores](#)[Manage Fines](#)[House Roster](#)[Administrative Tools](#)

### House Roster

Last Name	First	Email	Room Number
Agachenko	Sonia	sonia@berkeley.edu	21
Callaghan	Catherine	cat.callaghan@gmail.com	23
Cheng	Gong	travisinsanfran@gmail.com	4
Hartford	James	jmhartford@yahoo.com	10
Lui	Benjamin	benjamin.cy.lui@gmail.com	2
Maissy	Brian	brian.maissy@gmail.com	14
Morales	George	gmorales@berkeley.edu	29
Smith	John	johnhsmith@hotmail.com	8
Zahara	Eli	lemoncloud@gmail.com	16

#### House Roster View mockup (added in version 1.1)

When the WSM logs into the house roster, they will see a listing of the house residents. In the page, there are links (not pictured) that allow the WSM to import the information of the new residents from the CO, link to adding the residents, link to deleting residents, link to viewing one particular resident's workshift history, link to making a WSM account, link to resetting the password and link to editing one particular resident's background information.

## Configuring Chores/Shifts

Berkeley Student Cooperative Workshift Manager

[Home](#) [Profile](#) [Logout](#)

**STUDENT**  
**8op**  
**BERKELEY**

**Workshift**  
**Manager**

My Shifts

Shift Preferences

My Fines

Allocate Shifts

Monitor Shifts

Chores

Manage Fines

House Roster

Administrative Tools

### Chores

#### Create a chore

Name

Description

Time intervals

Days

☐ Monday ☐ Tuesday ☐ Wednesday  
☐ Thursday ☐ Friday ☐ Saturday ☐ Sunday

Number of Hours

Sign out deadline

Create Chore

#### Created Chores

Label 1
Content 1
Label 2

### Chores View mockup (added in version 1.1)

On the WSM's homepage, a link to configuring chores/shifts will be present. On that page (initially no chores or shifts present), there will be an option to create chores. Upon choosing this option, a form will allow the WSM to input a name, description, number of hours, and how long in advance a sign out is required. Once a chore is created, options to create a shift (instance of a chore), edit the chore, or delete the chore will be made available. To create a shift, the WSM will select an option next to a created chore, then input a specific date and time for the chore. Then the shift will be created and displayed underneath the respective chore. Also, a master list of chores with its respective shifts will be displayed on this page.



## Preference Form

Berkeley Student Cooperative Workshift Manager

[Home](#) [Profile](#) [Logout](#)

**STUDENT**  
**80p**  
**BERKELEY**

**Workshift**  
**Manager**

My Shifts

Shift Preferences

My Fines

Allocate Shifts

Monitor Shifts

Chores

Manage Fines

House Roster

Administrative Tools

**Preference Form**

Shift	Hours	Time Slot	Rating
Clean 1 <sup>st</sup> floor bathroom	1.5	None	
Clean 2 <sup>nd</sup> floor bathroom	1.5	None	
Clean 3 <sup>rd</sup> floor bathroom	1.5	None	
Dishes after breakfast	2	11am – 1pm	
Dishes after lunch	2	2pm – 4pm	
Dishes after dinner	2	9pm – 11pm	

Love it, want to do

Like it

Neutral

Don't like it

Absolutely hate or cannot do

Neutral

Save preferences

### Preferences View mockup (added in version 1.1)

Residents will be able to indicate their chore preferences using a form that displays all the chores and lets the residents give each chore a rating. Chores are displayed with the number of hours they are worth and the time slot they fall into, if any. Some chores, like washing the dinner dishes, must be done at a certain time, e.g. 9pm - 11pm, so the form will indicate that time slot for the dinner dishes shift. Other chores, such as cleaning the bathroom, do not have a specific time slot but must simply be done before the end of the week, so their time slot will be marked as “none” in the form. Next to each chore listing will be a drop-down menu from which the residents can select their rating for how much they like the chore. There are five ratings: “Absolutely hate or cannot do”, “Don’t like it”, “Neutral”, “Like it”, “Love it, want to do”. Once residents have marked their preference for each chore, they click on the “Save preferences” button at the bottom of the form to record the info they have just entered.

## Allocate Shifts

Berkeley Student Cooperative Workshift Manager

[Home](#) [Profile](#) [Logout](#)

STUDENT  
80p  
BERKELEY

Workshift  
Manager

My Shifts

Shift Preferences

My Fines

Allocate Shifts

Monitor Shifts

Chores

Manage Fines

House Roster

Administrative Tools

Shift Allocation

Daily

Weekly

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	
Cook Breakfast 7am - 9am	Lui, Ben	Hannigan, Lisa	Lui, Ben	Hannigan, Lisa	Lui, Ben	Maissy, Brian	Doe, Jane	Maissy, Brian 3/5hrs allocated
Breakfast Dishes 9am - 11am	Cheng, Travis	Smith, John	Cheng, Travis	Smith, John	Cheng, Travis			Callaghan, Cat 2/5hrs allocated
Cook Lunch 10am - 12pm								Lui, Ben 3/5hrs allocated
	Schuller, Tai	Callaghan, Cat	Callaghan, Cat	Callaghan, Cat	Schuller, Tai	Hannigan, Lisa	Hannigan, Lisa	Schuller, Tai 1/5hrs allocated
								Cheng, Travis 4/5hrs allocated
								Doe, Jane 4/5hrs allocated
								Smith, John 4/5hrs allocated
								Hannigan, Lisa 2/3hrs allocated


### Shift Allocation View mockup (added in version 1.1)

The shift allocation interface shows a list of residents (and the number of hours assigned to them) and a list of shifts (and the number of hours they take), side by side. The residents are ordered by increasing number of hours assigned, and the shifts by availability, chore, and date/time. When the user clicks on one of the residents, the shifts are rearranged in order of decreasing preference. When the user clicks on a shift, the residents are rearranged in order of decreasing preference for the selected shift. Residents who already have enough hours are grayed out but still shown in order. To assign a shift to a resident, the user drags the shift to the user or vice versa. Then the resident's name will appear under the shift and the resident's assigned hours will be increased appropriately.

## Manage Shifts

Berkeley Student Cooperative Workshift Manager

[Home](#) [Profile](#) [Logout](#)

**Workshift  
Manager**

[My Shifts](#)  
[Shift Preferences](#)  
[My Fines](#)  
[Allocate Shifts](#)  
[Monitor Shifts](#)  
[Chores](#)  
[Manage Fines](#)  
[House Roster](#)  
[Administrative Tools](#)

### Manage Workshifts

Day	Workshift	Resident	Status
Monday	Post-dinner pot	Mike	Completed
	Post-dinner dishes	Kenny	Pending
Tuesday	Post-dinner dishes	Cartman	Blown Off
	Post-dinner pot		Available

### Manage Shifts View mockup (added in version 1.1)

The shift management interface shows a list of all shifts, their date/time, their hour value, who they are assigned to, and their status. Using simple drop down menus, the user can change the status of a shift, unassign it from a resident, or assign an available shift to a resident. Clicking on any field of an existing shift allows the user to edit it in place, and clicking on the “Create new shift” text at the bottom prompts the user for a chore to base the shift off, and then adds it to the list with a blank date/time field.

## **Security Requirements**

Security is not a major issue in our system, because none of the data is particularly sensitive or valuable, so there is little motivation to attack the system and little cost if it is attacked. Nevertheless, we will build the system as secure as possible without incurring unreasonable overhead. Each user will have a username and a password with which to authenticate, and passwords will be encrypted before being sent over the network. Any web application framework should provide sufficient security by default. Similarly, there shouldn't be any motivation for DOS attacks.

## **System Requirements**

WSMA will be a web application. It will require a web server, an interpreter for the programming language we choose, and an installation of the web development framework we choose.

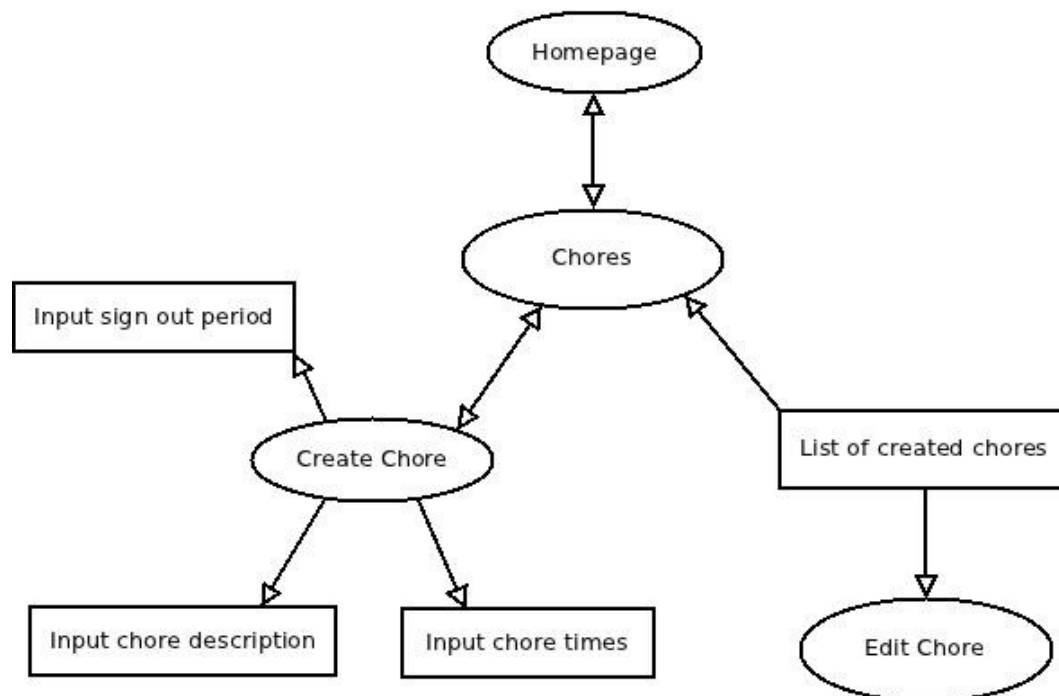
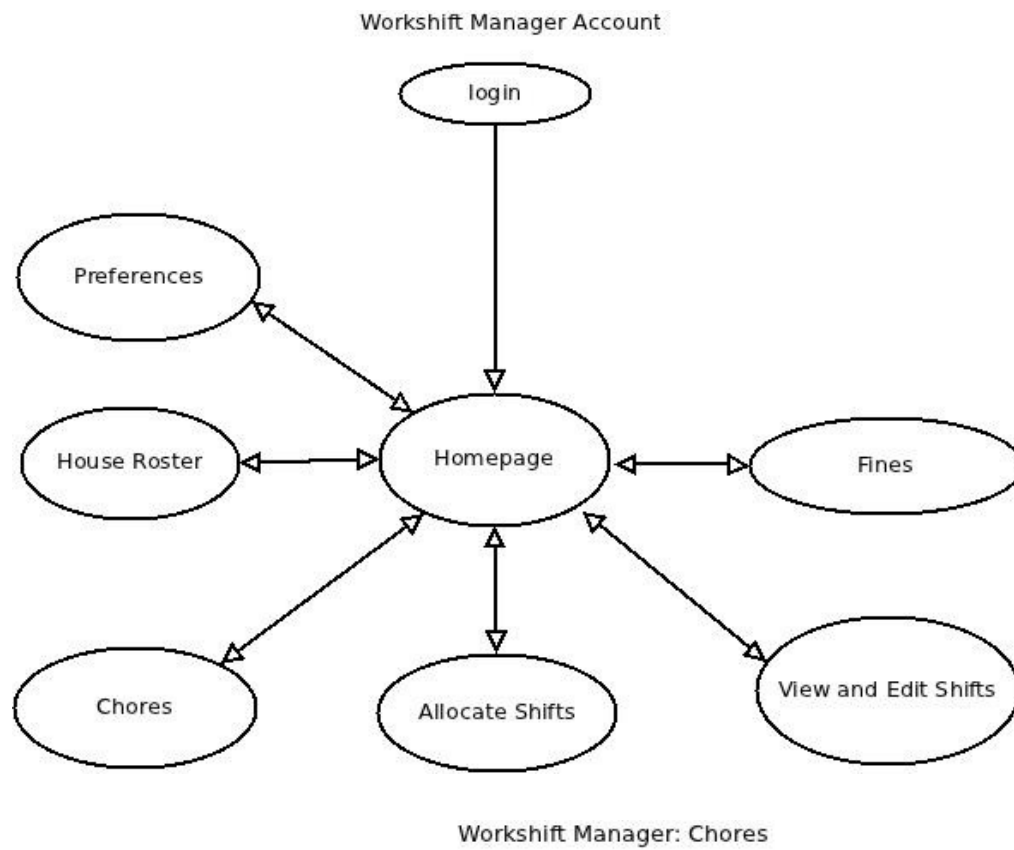
We will ensure that the application is compatible with all major browsers (firefox, ie, chrome, safari).

The application will need to interact with a database, use some utility to send emails, be able to import a list of residents as a csv file (or similar), export fine records as a csv file (or similar), and read data from a twitter feed. Also, as an extra feature, it will interact with the JomSocial API, to log into the WSMA with a JomSocial account.

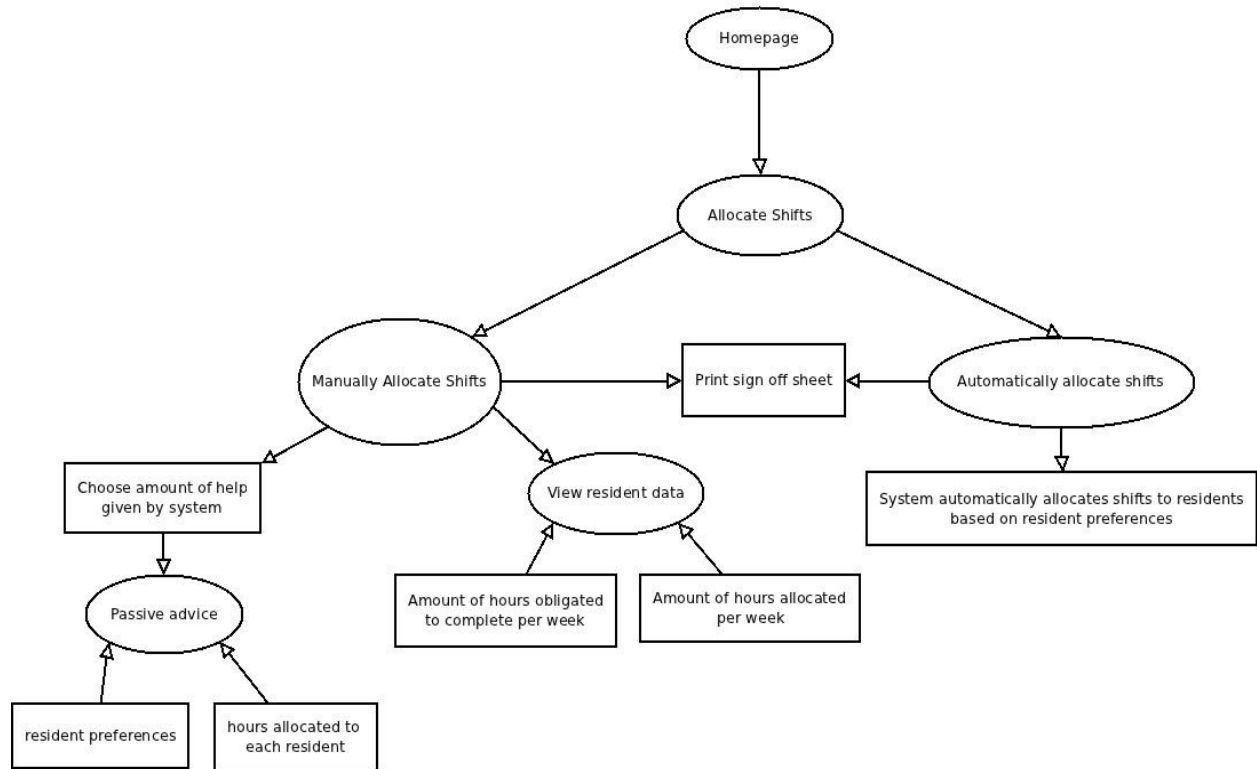
The system will also need to be able to schedule tasks to run later, such as calculating fines at scheduled intervals and marking shifts as blown off if they are not signed off by a certain deadline. These tasks do not run in the context of a client request / server response cycle.

The management of workshifts is neither data-heavy nor processing-intensive, so the system will have no particularly strenuous processing, memory, or storage requirements.

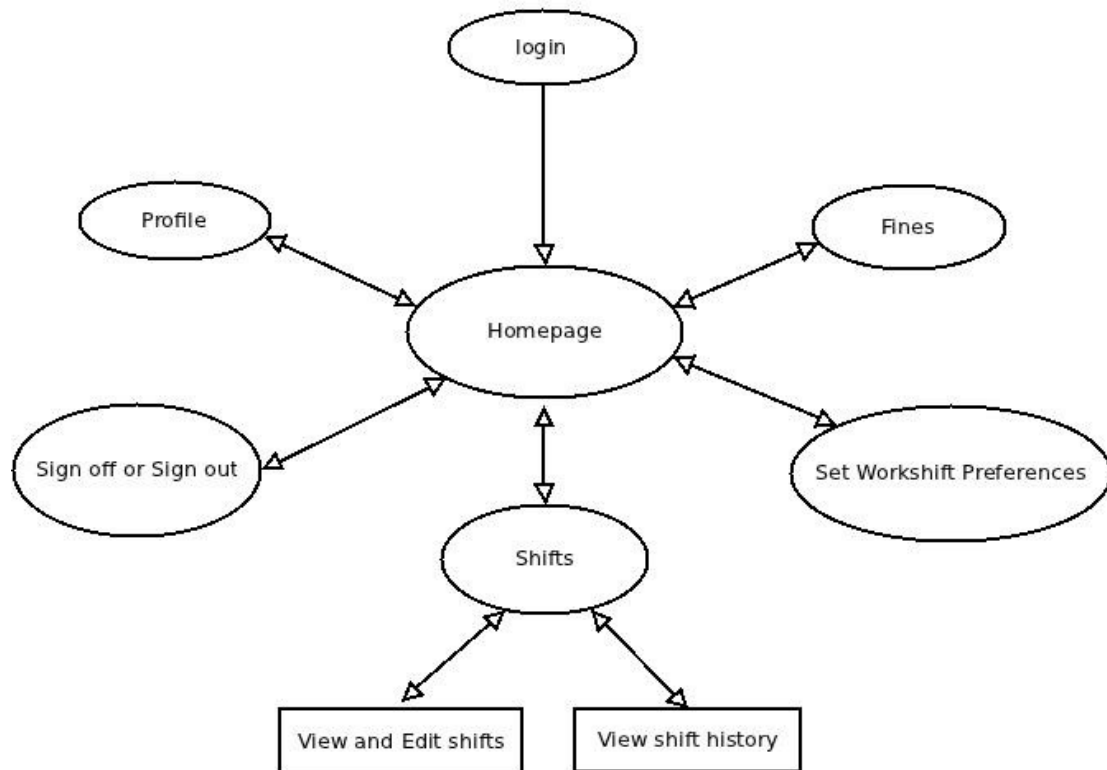
## Specification



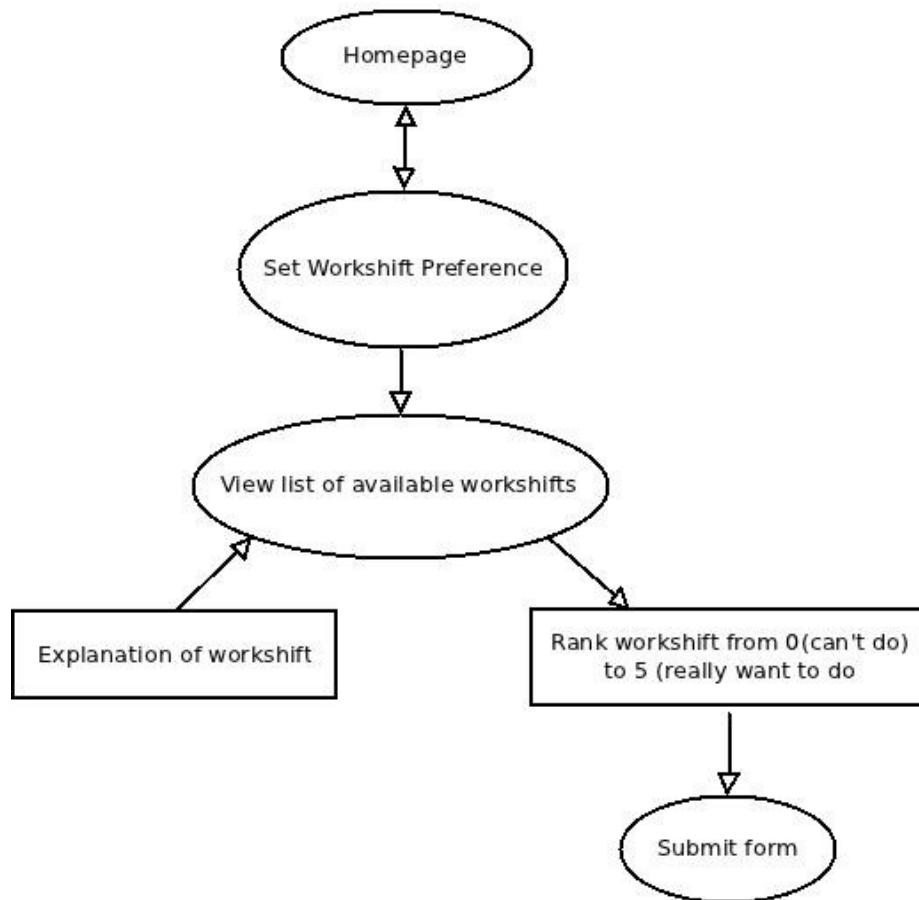
# Allocating Shifts to residents



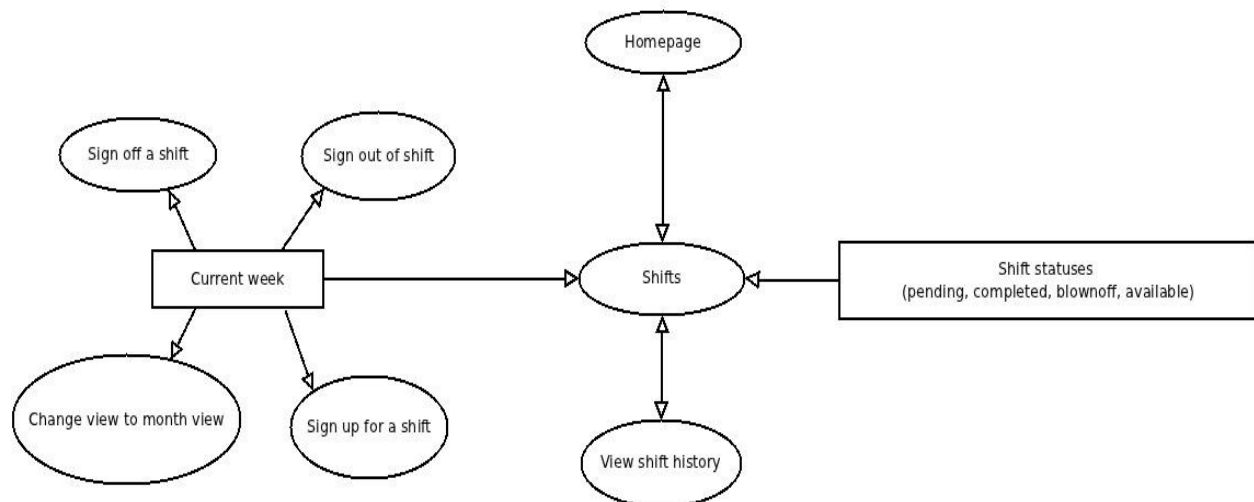
# Resident Account



Resident account: Set shift preference



Resident Account: Managing Shifts



## **Terminology**

- **WSMA: Workshift Manager's assistant.**
- **The system: WSMA.**
- **WSM: workshift manager.**
- **BSC: Berkeley Student Cooperative.**
- **House: one coop in the BSC.**
- **CO: the BSC central office.**
- **Resident: someone who lives in a house.**
- **Chore: a type of work (e.g. cook dinner) with associated data (e.g. takes 3 hours).**
- **Shift: a chore with an associated date/time (e.g. cook dinner Wednesday at 3).**
- **Allocated shift: shift WSM associates with a resident for the entire semester.**
- **Assigned shift: associating a shift with a resident for a specific week, happens automatically at the beginning of each week if the shift is allocated to the resident.**
- **Sign up: assign a shift to a resident.**
- **Sign out: unassign a shift to a resident. A resident can do this to indicate that he or she is unable to complete an assigned shift.**
- **Sign off: indicate that a resident has completed a shift.**
- **Blow off: not completing an assigned shift by the due date and time without signing out in advance.**
- **Optional: a feature that the WSM can turn on or off.**
- **Extra feature: a feature that we will implement if we have time.**