Selected Scenarios for HRIS

Use Case #8: User_views_StaffList

Requirement:

The system needs to be able to generate an interactive list of staff employed by the School.

Overview:

The user views the staff list when the application first loads or when switching to the staff list view. This list should display names in the format 'Family Name, Given Name (Title)', as in 'Einstein, Albert (Dr)', and be ordered alphabetically by family name. In addition to using the list to access staff details, the user can also filter the list, either by a staff member's employment category or (if time available in project) partial name match, or both. For basic filtering, the user should be able to list all staff, or just those who are academic, technical, administrative, or casual. If implementing name matching then the list contents would be restricted to show only those staff whose given name or family name contains the text entered by the user. Partial matches that span combinations of given and family name do not need to be supported, so look for matches within both names independently.

Preconditions:

1. HRIS database exists and is populated with data

Scenario:

Action	Software Reaction
1. User starts application	1. System creates MainView with StaffTab visible, which contains the StaffListView and StaffDetailsView 2. System retrieves basic staff details from database: name, title and category 3. Staff are sorted alphabetically by family name 4. System shows names in StaffListView
2. User selects staff tab	1. System shows staff tab
3. User selects a staff category (including the 'all staff' category) from a dropdown list of categories	1. System updates current set of filters to include selected category 2. System creates temporary list of staff based on current set of filters (category or name, if present) 3. System refreshes the StaffListView with the filtered list
3. User modifies text in filter-by-name text field	1. No reaction; not supporting automatic refresh of list contents
4. User presses enter key inside filter-by-name text field	1. System updates current set of filters to include given text or, if search text is blank, removes any existing text from set of filters 2. System creates temporary list of staff based on current set of filters (category or name, if present) 3. System refreshes the StaffListView with the filtered list
5. User selects staff name in list	See *UC16_User_selects_StaffDetails*

Scenario Notes:

Actions 1 is a precondition for all others. The other actions may occur in any order, although it is more typical for Action 3 to be followed by Action 4 (the user pressing enter to trigger name-based filtering). The reaction to Actions 2 and 4 produces the intersection of the two filters.

Post Conditions

StaffListView displays all or a subset (particular category or with matching names) of staff names

Required Views: (GUIs)

MainView containing StaffTab (list + detail), and StaffListView

Exceptions:

1. If the database is inaccessible or another unexpected error occurs then the system should respond with a message to this effect, requesting that the user close and restart the software. (In a later release the system should be made robust to such exceptions and be able to reload the database without being restarted.)

Use Cases utilised

UC16_User_selects_StaffDetails

Use Case #16: User_selects_StaffDetails

Requirement:

When the users selects a name in the list the system will show more details about the staff member (referred to as the Staff Details view), which should include: Name; Campus; Phone Number; Room Location; Email Address; Photo; Consultation hours; Table of units he or she is involved with in the current semester.

Overview:

From either the staff list + detail or unit timetable displays the user can select a staff member to view their details (the second will be implemented if time allows). In the first case they select the staff member's name in the list, while in the second case they select a staff member in an entry in the timetable.

In addition to the basic details about each staff member, if development time allows then staff member's current availability should also be displayed: 'teaching' (with details of the unit code and room) if they are in a timetabled class; 'consulting' if it is during their consultation times; and 'free' otherwise.

Preconditions:

1. HRIS database exists and is populated with data

Scenario:

Action	Software Reaction
1. User selects staff name in StaffListView	1. System retrieves any additional details about the staff member not previously loaded 2. System determines availability status of staff member by comparing current system time with staff member's consultation times and timetabled classes he or she teaches 3. If not previously loaded system retrieves list of units the staff member is involved in teaching 4. System enlarges application window size if too small to display StaffDetailView. 5. System presents staff details and availability information in StaffDetailView
2. User selects staff name in timetable entry in UnitTimetableView	1. System switches to StaffTab 2. Remaining reactions as per response to action (1)
3. User clicks 'Activity Grid' button	See *UC19_User_shows_ActivityGrid*

Scenario Notes:

Actions 1 or 2 are independent triggers for viewing staff details.

Post Conditions

StaffDetailView displays details

Required Views: (GUIs)

StaffTab
StaffListView
StaffDetailsView
UnitTimetableView

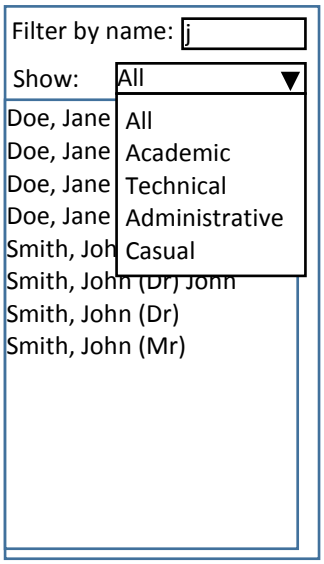
Exceptions:

1. If the database is inaccessible or another unexpected error occurs then the system should respond with a message to this effect, requesting that the user close and restart the software. (In a later release the system should be made robust to such exceptions and be able to reload the database without being restarted.)

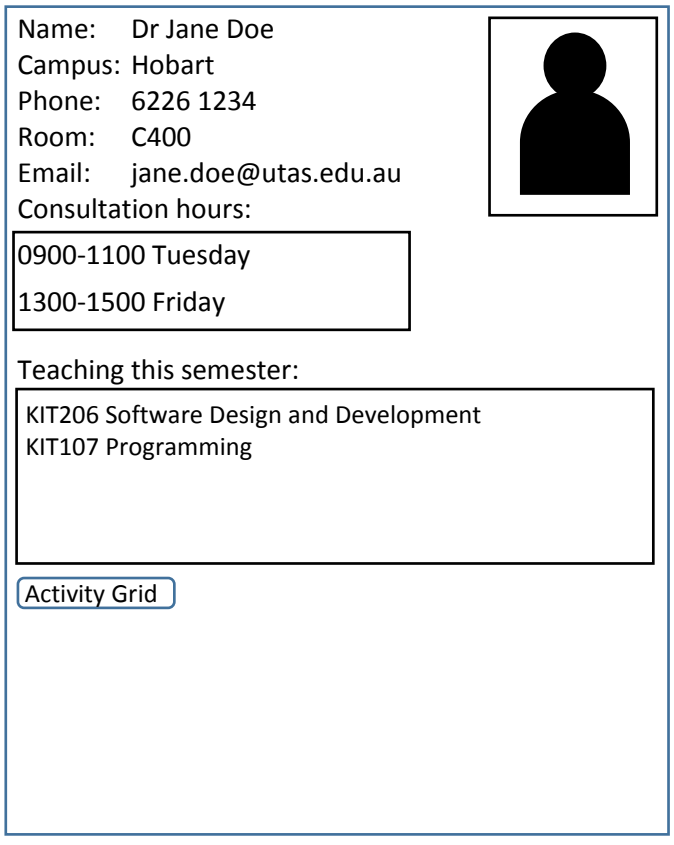
Use Cases utilised

UC8_User_views_StaffList (precondition for action 1)
UC19_User_shows_ActivityGrid

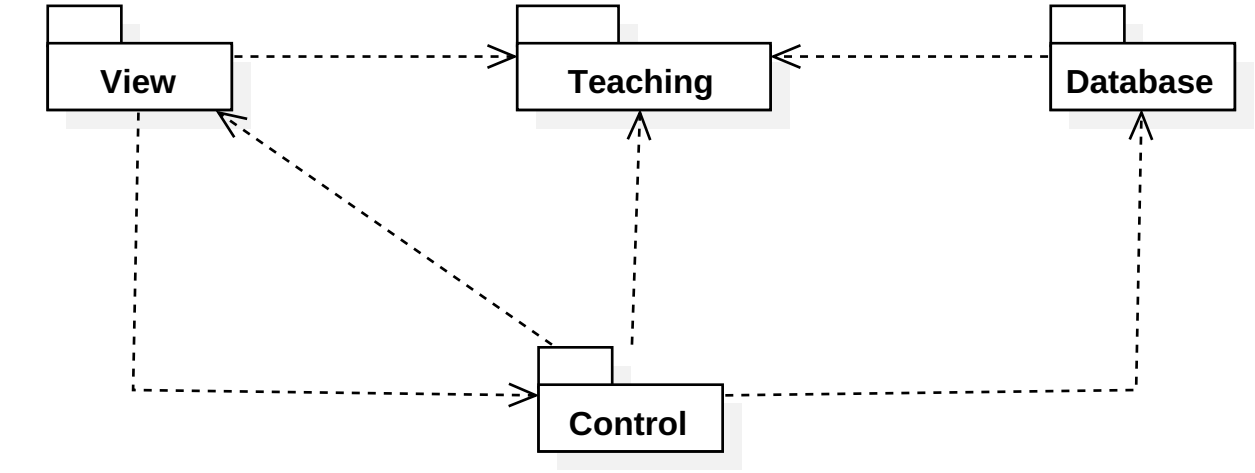
Example low fidelity prototypes
These represent one possible way of presenting parts of the HRIS application.



StaffListView

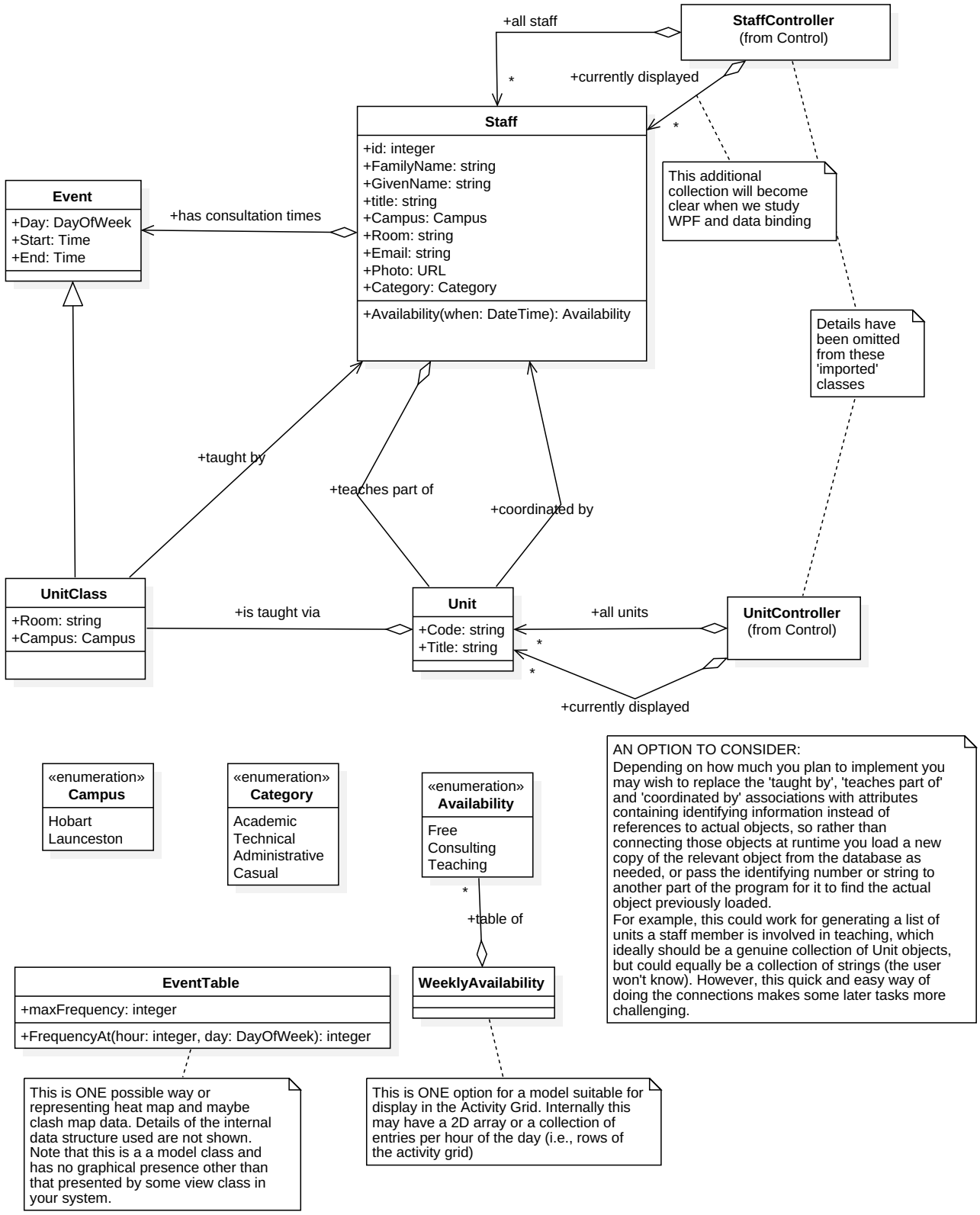


StaffDetailsView

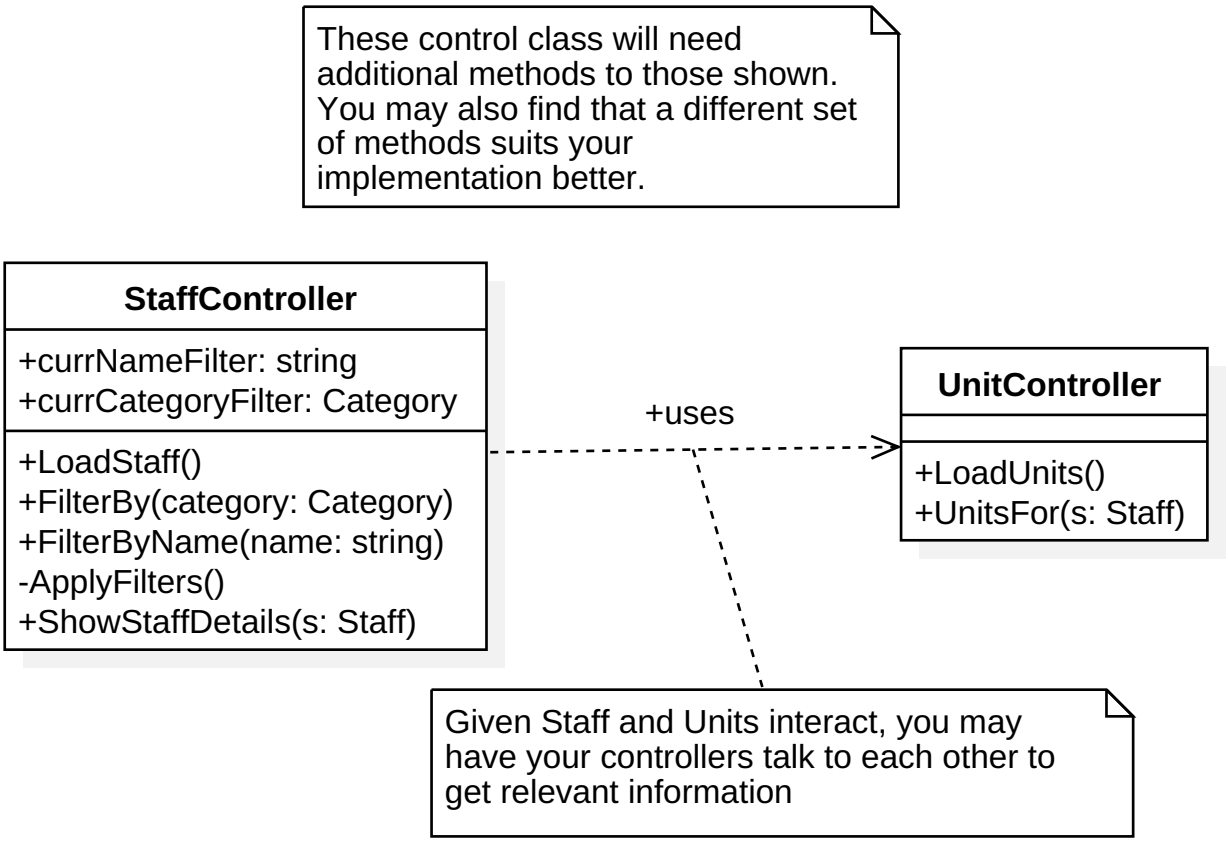


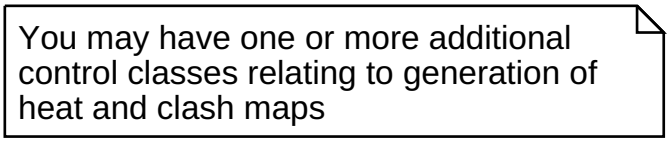
The bidirectional dependency is because views pass messages back to controllers

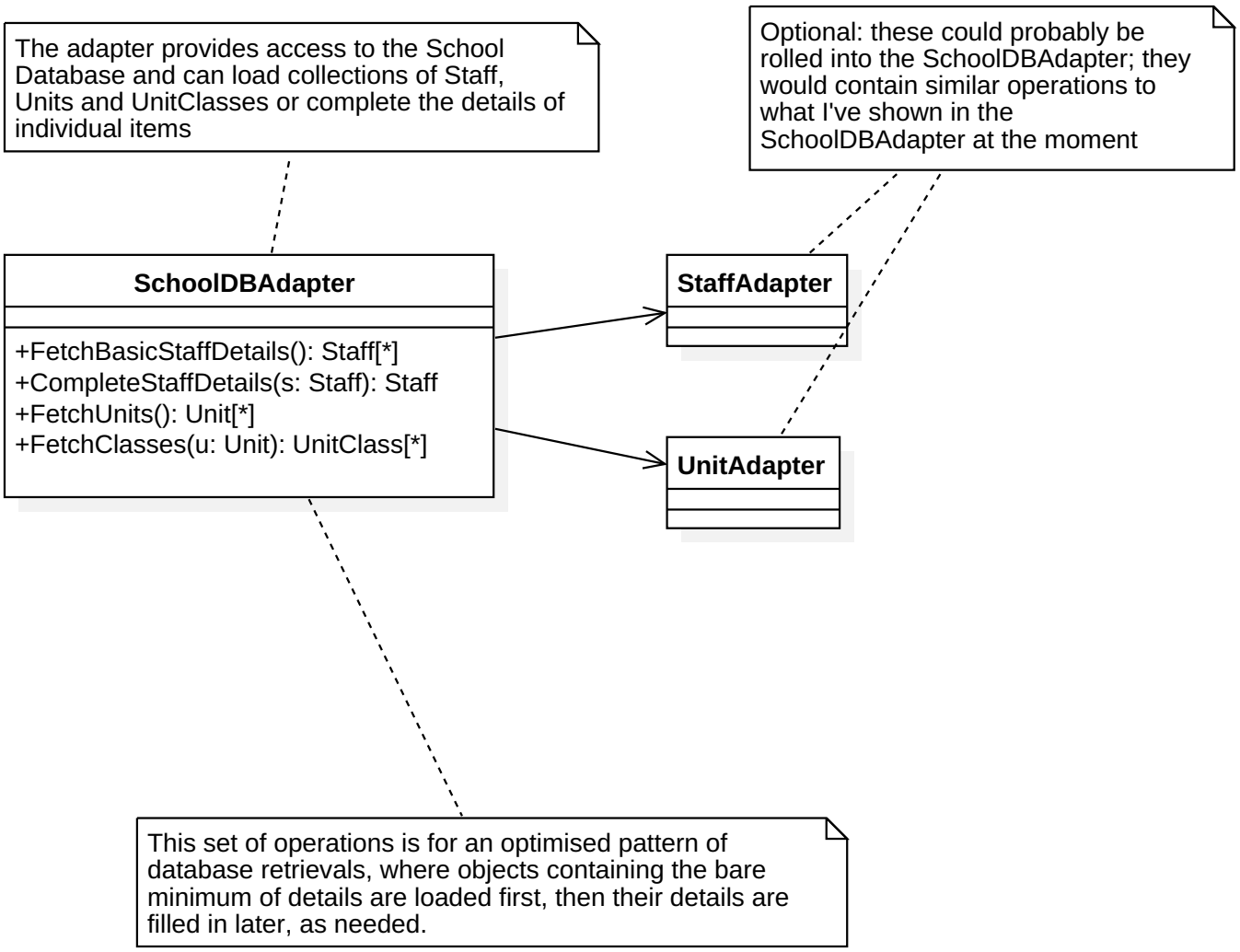
HRIS models::Teaching::Teaching classes



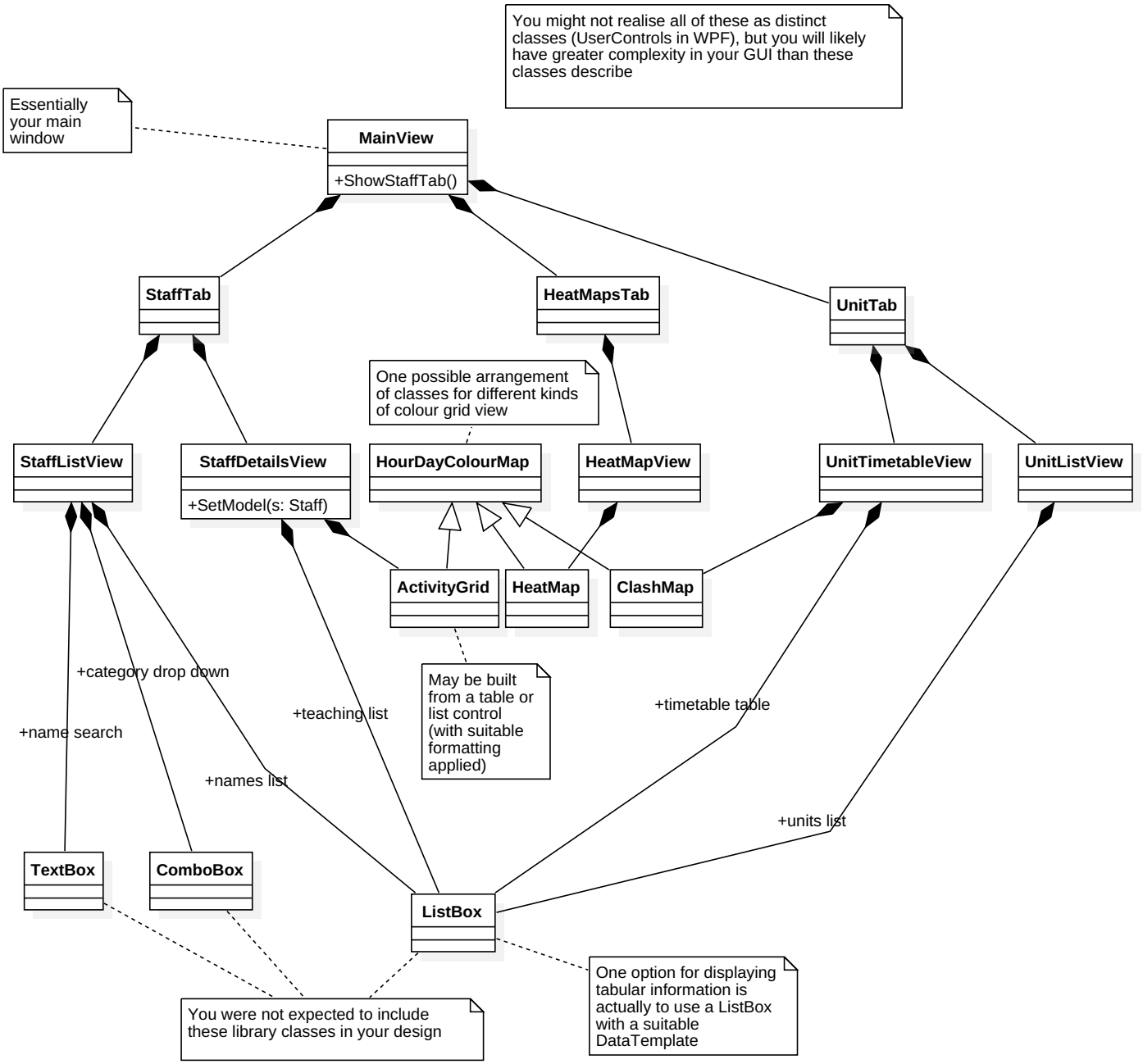
REWORK THIS TEXT: In an ideal OO model (whch is what we were looking for in Assignment 1) there would be
associations between Staff and Unit and UnitClass (you may have called yours Lesson).
To enable independent loading of entities from the database these connections have been replaced in this sample model with identifying information in attributes (each Unit has the integer ID of its coordinator, each UnitClass has the integer ID of the staff member teaching it, each Staff member has a collection of unit codes).
(In the future, when you use the Entity Framework to manage object persistence in a database, this will not be necessary.)


These control class will need additional methods to those shown. You may also find that a different set of methods suits your implementation better.
StaffController
+currNameFilter: string
+currCategoryFilter: Category
+LoadStaff()
+FilterBy(category: Category)
+FilterByName(name: string)
-ApplyFilters()
+ShowStaffDetails(s: Staff)
+uses
UnitController
+LoadUnits()
+UnitsFor(s: Staff)
Given Staff and Units interact, you may have your controllers talk to each other to get relevant information

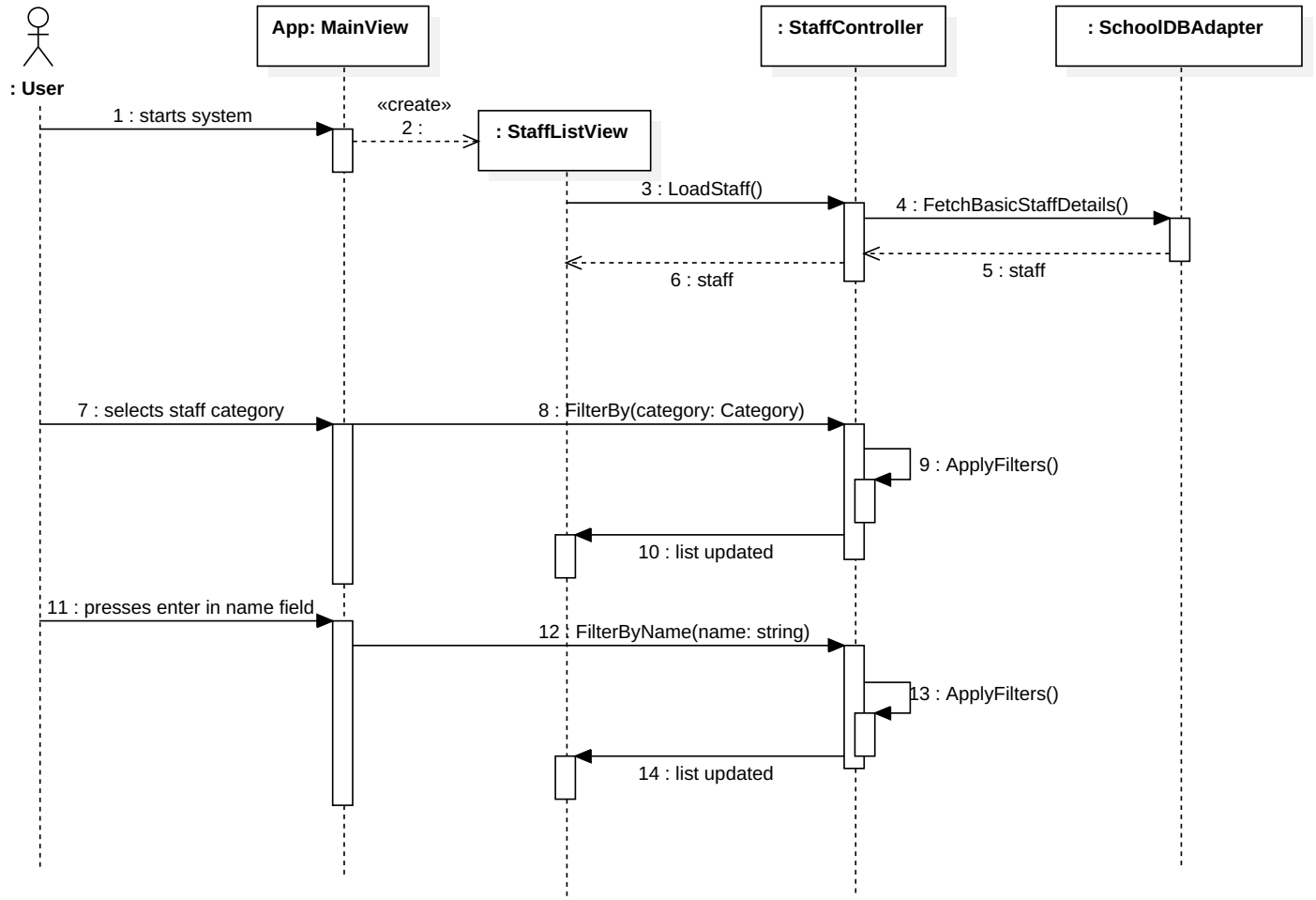

You may have one or more additional control classes relating to generation of heat and clash maps


The adapter provides access to the School Database and can load collections of Staff, Units and UnitClasses or complete the details of individual items
Optional: these could probably be rolled into the SchoolDBAdapter; they would contain similar operations to what I've shown in the SchoolDBAdapter at the moment
SchoolDBAdapter
+FetchBasicStaffDetails(): Staff[*]
+CompleteStaffDetails(s: Staff): Staff
+FetchUnits(): Unit[*]
+FetchClasses(u: Unit): UnitClass[*]
StaffAdapter
UnitAdapter
This set of operations is for an optimised pattern of database retrievals, where objects containing the bare minimum of details are loaded first, then their details are filled in later, as needed.

HRIS models::View::View classes



HRIS models::Sequence diagrams::UC8_User_views_Staff::UC8 sequence diagram



HRIS models::Sequence diagrams::UC16_User_selects_StaffDetails::UC16 sequence diagram