berryLife
Playbook Personal Scheduling Application

Eric Singh

Table of Contents

Table of Figures

Industry Overview

There are a variety of personal organization and scheduling application available for all platforms. Most of these are fairly simplistic and targeted mainly towards personal use. While these apps are similar to BerryLife it differentiates itself by allowing users to improve their scheduling practices by integrating GPS functionality. Such feature being targeted to both personal and business users.

Product Overview

The BerryLife is a personal organization and scheduling application. It's main features are as follows:

- Creating/editing ToDo lists and Events which can be added to the devices calendar
- Saving physical locations using GPS which can be associated with created/existing events and to-do lists.

The application itself is separated into three main areas: the Working Area, Menu and the Sidebar

The Working Area is where the user will make more of their interactions with the application. It serves as a hub for all tools included in the application (calendar, creation of Events/ToDos/GPS locations, editing, viewing indexes).

The Menu is overlaid on the Working Area. It consists of four corner buttons that when press expand to make relevant sub-options available to the user. Three of the buttons are contextual and one is fixed. The contextual buttons change based upon the tool currently being used in the Working Area, whereas the fixed button always contains options pertaining to application preferences.

The Sidebar is a dynamic area where data can be presented to the user in an easy to understand format. "Active" Events and ToDos are listed here for easy reference.

The calendar portion of the application will function as a normal calendar with views ranging from monthly to hourly. Activities/events can be set in any view and will cascade downwards but only be viewable at the appropriate level of view. For example: If an event were to be placed on a month, it would cascade downwards to all the days/hours included in that month, but the event itself would only be visible at the monthly view. This data is extracted from/inserted into the device's calendar application.

Events are text elements describing a task/appointment/etc. that require a time period and advance notification time upon creation. An event that has reached its notification time will notify the user and appear as "active" in the Sidebar. Events may also be associated with a saved GPS location. If the user is within the defined range around the GPS location the event will notify the user and appear as "active" in the Sidebar.

ToDos are text list elements that outline a set of tasks that may or may not have a time period associated with them. If a ToDo is not associated with a time period it will be "active" and visible in the Sidebar until the ToDo is deemed completed by the user. A ToDo that is associated with a time period will notify the

user and appear “active” in the Sidebar during the selected time period. ToDos may also be associated with saved GPS locations. If the user is within the defined range around the GPS location the event will notify the user and appear as “active” in the Sidebar.

Actors and Scenarios

Actors

There will be only one User group in the application. This group consists of the individual who has the app installed on their Blackberry Playbook and any secondary users. The main User of the app will have the ability to execute all of the application’s main functions and secondary functions.

Scenarios

new_GPS_location

Participants:
User
Flow:
1. User is at, or has the lat-long coordinates for, a location that they wish to tag an event or to-do list. User uses the “new location” function.
2. User enters the lat-long coordinates if required and a distance around the coordinates for which the location is active. User confirms this information.
3. Once the location is verified, User is notified that the new location has been stored and is taken back to their home screen

midterm

Participants:
User
Flow:
1. User has a midterm coming up and would like to set up an event for the time of the midterm. User selects the time of the midterm and uses the “new event” function to set up an event for it.
2. User fills out the information for the midterm and sets a time before the event to be notified.
3. Once confirmed, User is taken back to home screen where the event is visible on the calendar and they will be notified at the time before the event that was set.

to_do_onday

Participants:
User
Flow:
1.User knows they need to complete a set of tasks in a particular time in the future. They use the “create new to-do” function to make up a list of those tasks.

2. User fills out the list and then selects a time period they want to attach this list to.
3. User uses the “chronological association” function to attach the list to selected time period.
4. Once confirmed, User is taken to their home screen where the to-do list is visible within the time period and the User will be notified of the list once they enter into that time period.

grocery_list

Participants:
User
Flow:
1. User is going shopping later and needs to write a grocery list for when they go to the store. They use the “create new to-do” function to make up a grocery list.
2. User fills out the list and decides that this list does not need to be attached to a date or time, but instead to the GPS location they have saved for the store previously.
3. Either during or after the creation of their list User uses the ‘GPS tag to-do” function to associate it with the location of the store.
4. Once confirmed the User is taken back to their home screen where they will be notified of their to-do list when they enter the proximity of the grocery store set earlier.

library_due dates

Participants:
User
Flow:
1. User frequents their public library and has made events for each book’s due date using the “new event’ function. They also used the “GPS tag event” function to tag each event to the public library.
2. Assuming all events and GPS tags were set up correctly the User will be notified of all due books when entering the library.

Initial Use Case Identification

Use Cases

---

tagEvent

Participants:
User
Entry condition:
An event exists to be tagged
Flow:
1) User selects the event that he wishes to tag a GPS location to

2) On the selected event User will press button “Tag to Location”
3) App will display a menu asking weather to add a new location or use existing location
5) If the User selects to add a new location call “newLoc” case and return with the corresponding coordinates
6) If the User selects to use existing location a menu will appear presenting the list of saved locations which the User can select and returns the corresponding locations coordinates
6) The app will the link the coordinates to the corresponding event

Exit condition:

The User has successfully tagged a location to an event or has chosen to abort

Special requirements:

None

tagToDo

Participants:

User

Entry condition:

A to-do list exists to be tagged

Flow:

1) User selects the to-do list that he wishes to tag a GPS location to
2) On the selected event User will press button “Tag to Location”
3) App will display a menu asking weather to add a new location or use existing location
5) If the User selects to add a new location call “newLoc” case and return with the corresponding coordinates
6) If the User selects to use existing location a menu will appear presenting the list of saved locations which the User can select and returns the corresponding locations coordinates
6) The app will the link the coordinates to the corresponding to-do list

Exit condition:

The User has successfully tagged a location to a to-do list or has chosen to abort

Special requirements:

None

newEvent

Participants:

User

Entry condition:

No conditions.

Flow of events:

1) User chooses to create event
2) App displays menu buttons for the creation of new event.
3) User inputs event data
4) If User chooses to tag a GPS location to this event tagEvent is invoked
5) If User chooses to associate this event with a time period chronoEvent is invoked
6) Once the list is saved, the event waits to be deployed

Exit condition:

The User has successfully created new event or has aborted during the creation

Special requirements:

None

newToDo

Participats:

User

Entry condition:

No conditions.

Flow of events:

1) User chooses to create a to-do list
2) App displays menu buttons for the creation of new to-do list.
3) User inputs to-do list data
4) If User chooses to tag a GPS location to this to-do list tagToDo is invoked
5) If User chooses to associate a time period with this to-do list chronoToDo is invoked
4) Once the list is saved, the to-do list waits to be deployed

Exit condition:

The User has successfully created new to-do list or has aborted during the creation

Special requirements:

None

newLoc

Participants:

User

Entry condition:

GPS system can receive data from GPS satellites

Flow:

1) App requests if User would like to save their current GPS location
2) If User selects yes the User will also be required to enter in the name of the location and the radius to the User's current location
3) The app will then save the GPS coordinates of the location under a locations list that the User will be able to view and access

Exit condition:

The User has successfully added a new location to the locations list, the User selects "no" when asked if desired to add a new location, or the GPS cannot connect to a satellite

Special requirements:

None

chronoEvent

Participants:

User

Entry Conditions:

Creation of a new Event

Flow:

1) User creates a new Event
2) App prompts User to select a time period using calendar.
3) Once User confirms time period the event is associated with it.

Exit Conditions:

User successfully selects a time period to associate or chooses to abort the process.

Special Conditions:

None.

chronoToDo

Participants:

User

Entry Conditions:

Needs there to be a pre-existing, valid to-do list that does not already have a time period associated with it.

Flow:

1) User selects pre-existing event that has not been associated with a time perios form the list of to-do lists.
2) App prompts User to select a time period using calendar.
3) Once User confirms time period the to-do list is associated with it.

Exit Conditions:

User successfully selects a time period to associate or choose to abort the process.

Special Conditions:

None

Case Diagram

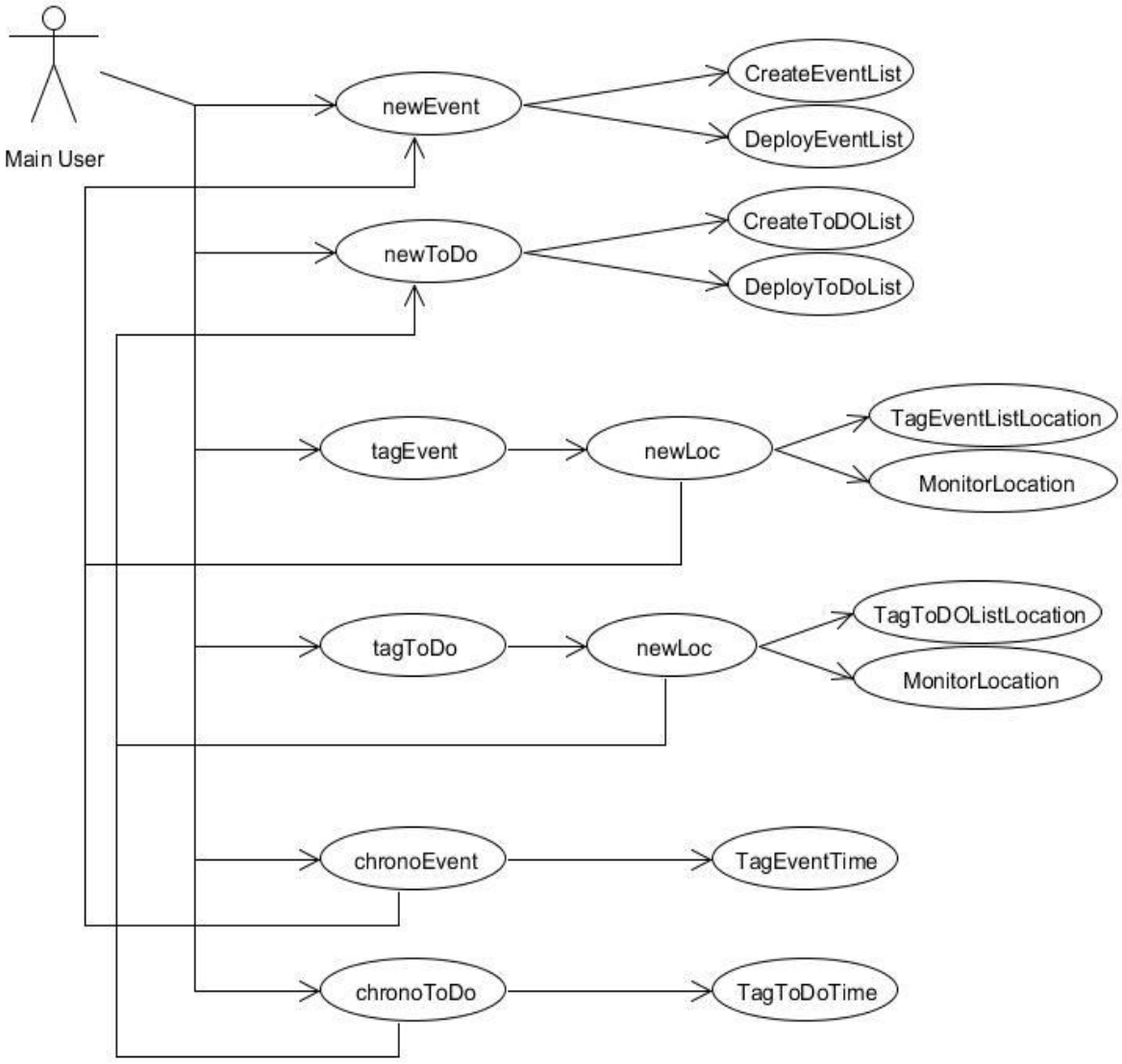


Figure 1

Nonfunctional requirements

1. Monitor will track current position and signal the application when User enters the GPS locations set out in tagEvent/tagToDo. It will also track the current time and signal the application at times set in chronoEvent/ChronoToDo
2. Confirmation of completion message appear on the screen when the chronoEvent or chronoToDo is due. Message stays on the screen until User responds to it. If the task is done, event will be taken off from the list. If the task is not done, event will be classified as an overdue event.
3. System shows User its overdue event on a daily bases, reminding User to either extend the event period or delete such event from the overdue list.

Objects

Entity Objects

Action

A generalized term for the common properties of events and to-do lists. This would include the ability to create time periods using the CalendarManager and associating with a GPSLocaion.

Event

A specific definition of an Action that requires an associated time period to function correctly. An Event would need a description of the event, a notification offset and the time period to be associated with it from User. Additionally, a GPSLocation could also be associated if User so chooses.

ToDo

A specific definition of an Action that does not require a time period or GPSLocation to function correctly. A ToDo would need a list of tasks to be completed only. Additionally, a time period or GPSLocation could also be associated if User so chooses.

GPSLocation

A GPSLocation contains data for a single location in the physical world.

Boundry Objects

WorkingArea

An area of the UI which User will be interacting with for the main part. This area can contain the CalendarManager, ToDoDex, EventDex, ActionGen or LocGen.

CalendarManager

A UI for the calendar potion of the application. Manages all calendar elements in the current ‘view’. (Ex. In ‘monthly’ view CalendarManager would allow access to Months, In ‘daily’ view Calendar Manager would allow access to Days)

Month

A UI element that is managed by the CalendarManager which represents a generic month of the year. Months contain Weeks and Days. Any Action applied to a Month will also be applied to the contained Weeks/Days.

Week

A UI element that is managed by the CalendarManager which represents a generic week in a month. Weeks contain 7 Days. Any Action applied to a Week will also be applied to the contained Days.

Day

A UI element that is managed by the CalendarManager which represents a generic Day in a month. Days contain 24 Hours. Any Action applied to a Day will also be applied to the contained Hours.

Hour

A UI element that is managed by the CalendarManager which represents a hour in a day.

ActionGen

A UI used during the creation/editing of Events or ToDos. This is where User enters a description, creates a task list, sets notification offset and chooses time periods/GPSLocations to associate with the created Event or ToDo.

LocGen

A UI used during the creation/editing of GPSLocations. This is where the user enters/captures GPS Data.

ToDoDex

A UI that displays a list of all ToDos that have been created by User. From this UI User can edit their ToDos using ActionGen.

EventDex

A UI that displays a list of all Events that have been created by User. From this UI User can edit their Events using ActionGen.

Menu

Four corner buttons within WorkingArea, each with a category, which extend to display options withing that category. These categories changed based upon the content of WorkingArea.

Menu Elements

The option buttons within each category of the Menu which perform specific functions.

Sidebar

A vertical bar to the left of the WorkingArea which will feature dynamic lists of SidebarElements based upon Actions which are currently ‘active’ . (Ex. an Event scheduled for today would be displayed here.)

Sidebar Elements

A UI element representing either an Event or ToDo to be displayed by the Sidebar. These will be distinguished using a graphic device (icon, background).

Control Objects

Monitor

A constantly running background process that will link the entity objects and boundary objects together. The monitor will keep track of the current Events, GPSLocations, ToDos and user input. The Monitor will also handle notifications.

EventManager

Keeps track of all Events created by User.

ToDoManager

Keeps track of all ToDos created by User.

LocationManager

Keeps track of all GPSLocations created by User. Has the ability to create new GPSLocations using device GPS capabilities.

Use Case/Object Map

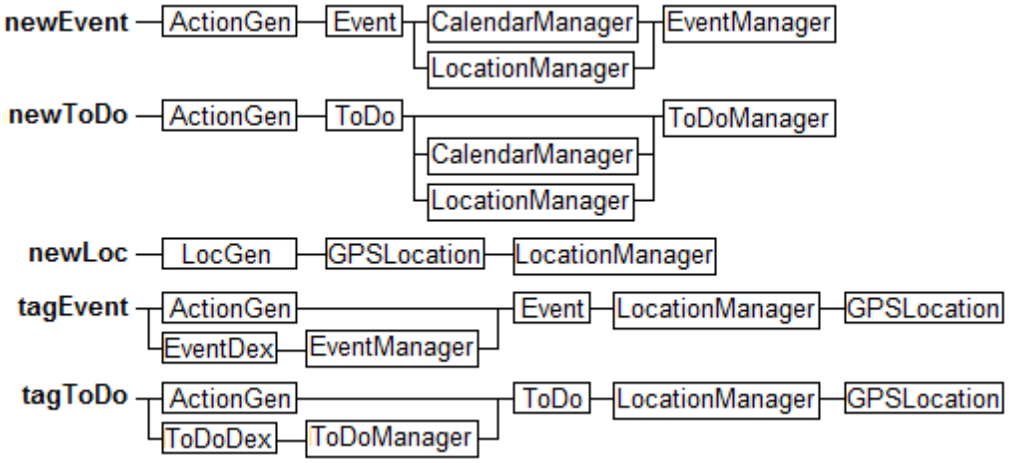


Figure 2

Identifying Associations Among Objects

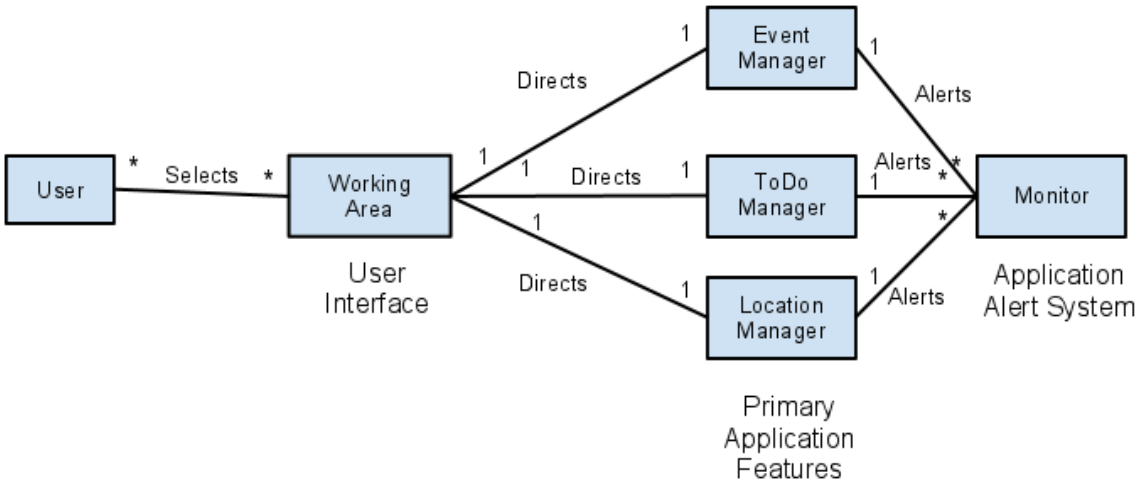


Figure 3

Identifying Objects Attributes

Action

Location: GPSLocation
TimePeriod: Month(s)/Week(s)/Day(s)/Hours(s)

Event

Description: string
NotificationOffset: integer(seconds)

ToDo

TaskList: string[]

GPSLocation

Location: GPS device data
Description: string

EventManager

EventArray: Event[]

ToDoManager

ToDoArray: ToDo[]

LocationManager

LocationArray: GPSLocation[]

Modeling nontrivial behavior with state charts

Event class:

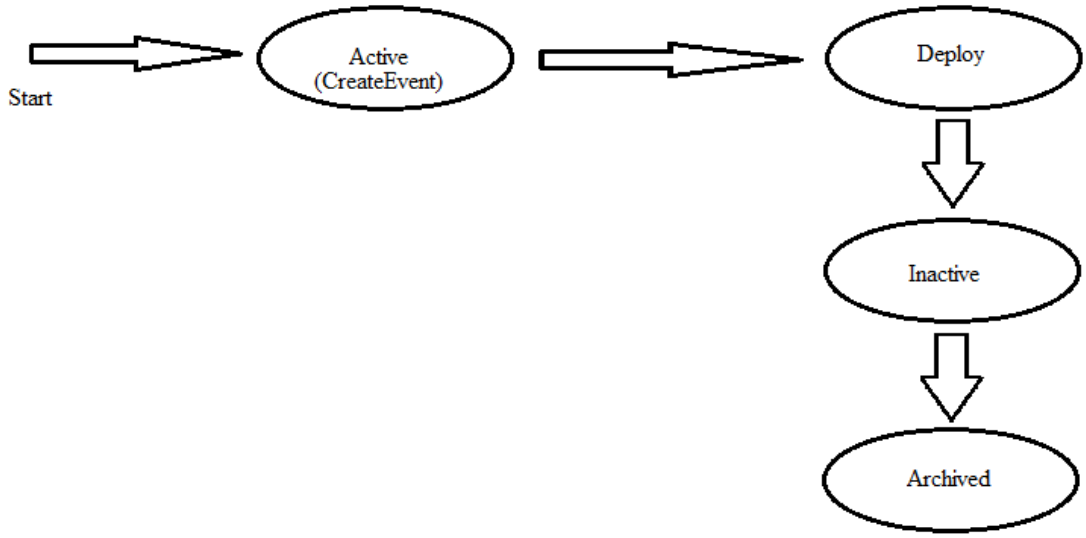


Figure 4

Todo class:

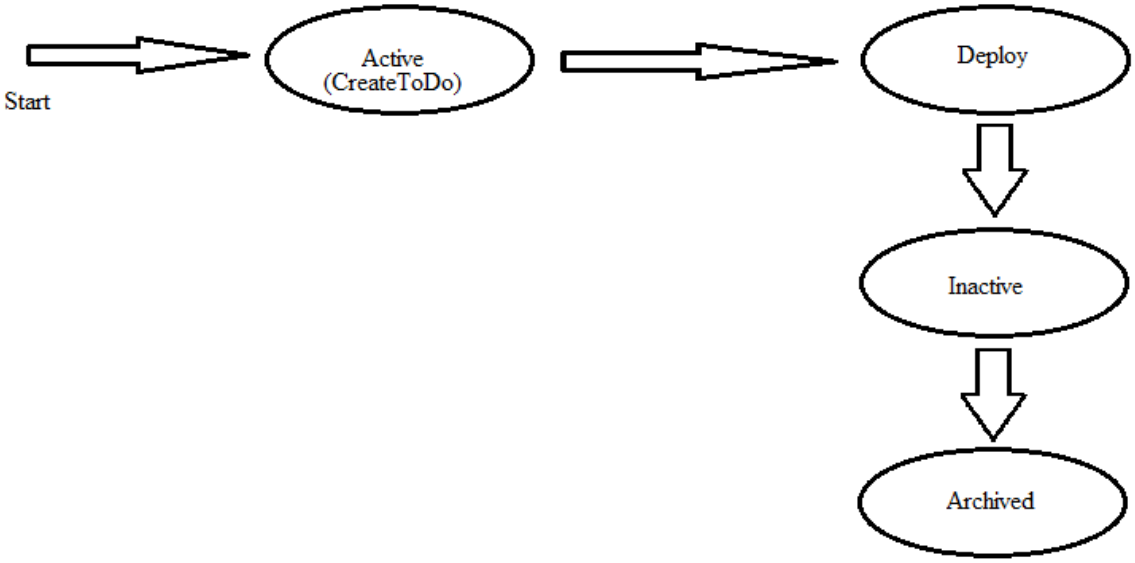


Figure 5

A sequence diagram for each use case

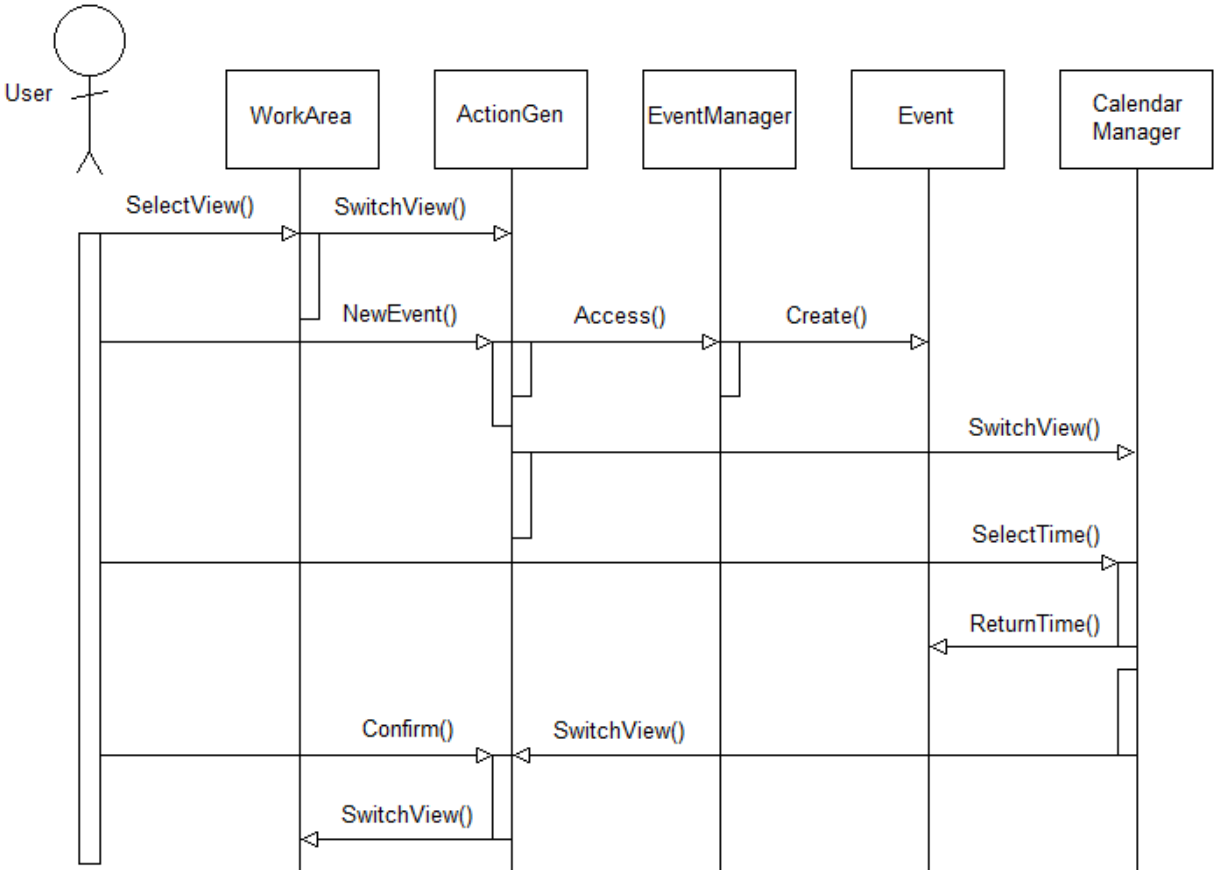


Figure 5: Sequence Diagram for newEvent

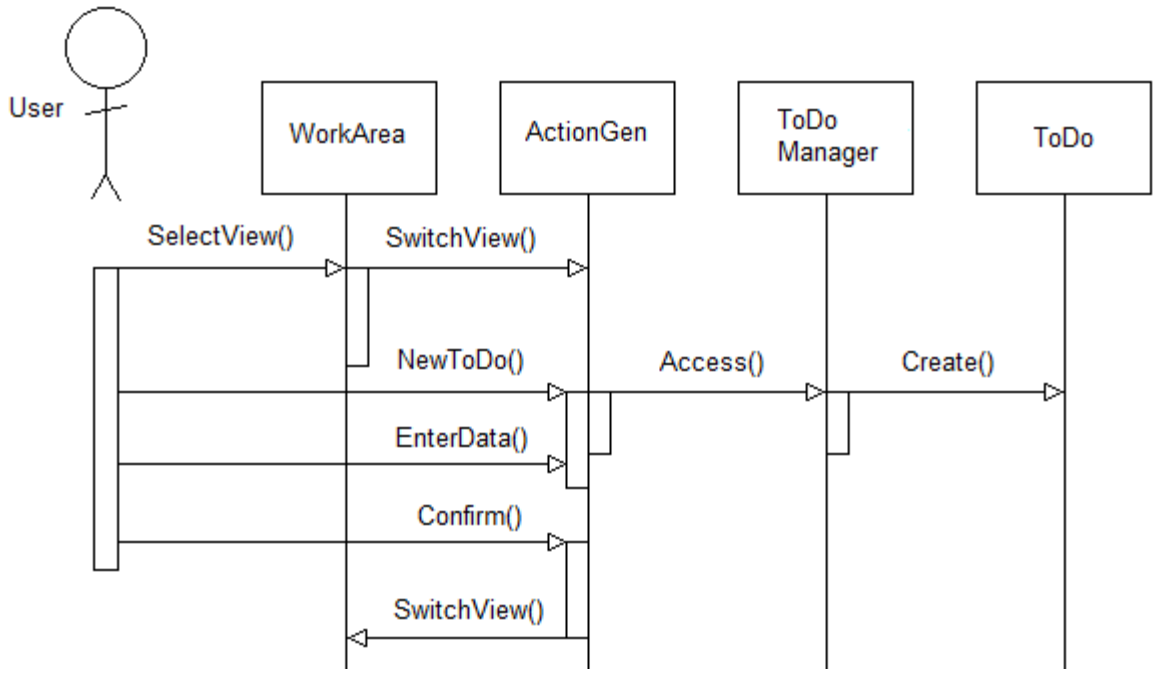


Figure 6: Sequence Diagram for newToDo

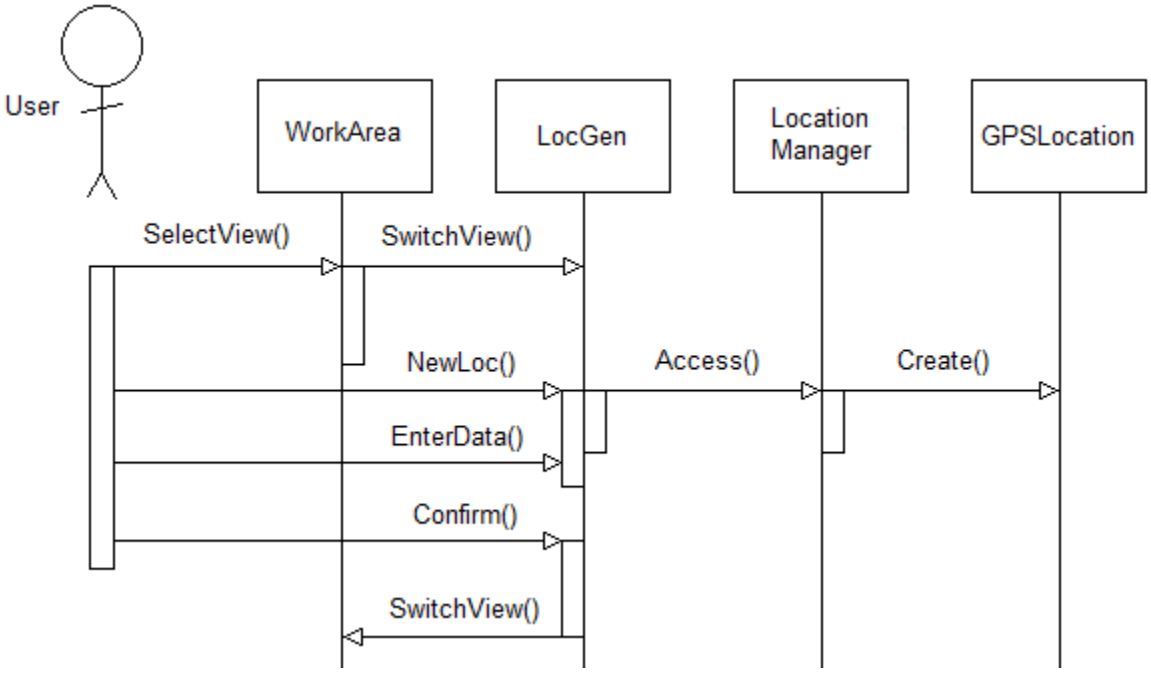


Figure 7: Sequence Diagram for newLoc

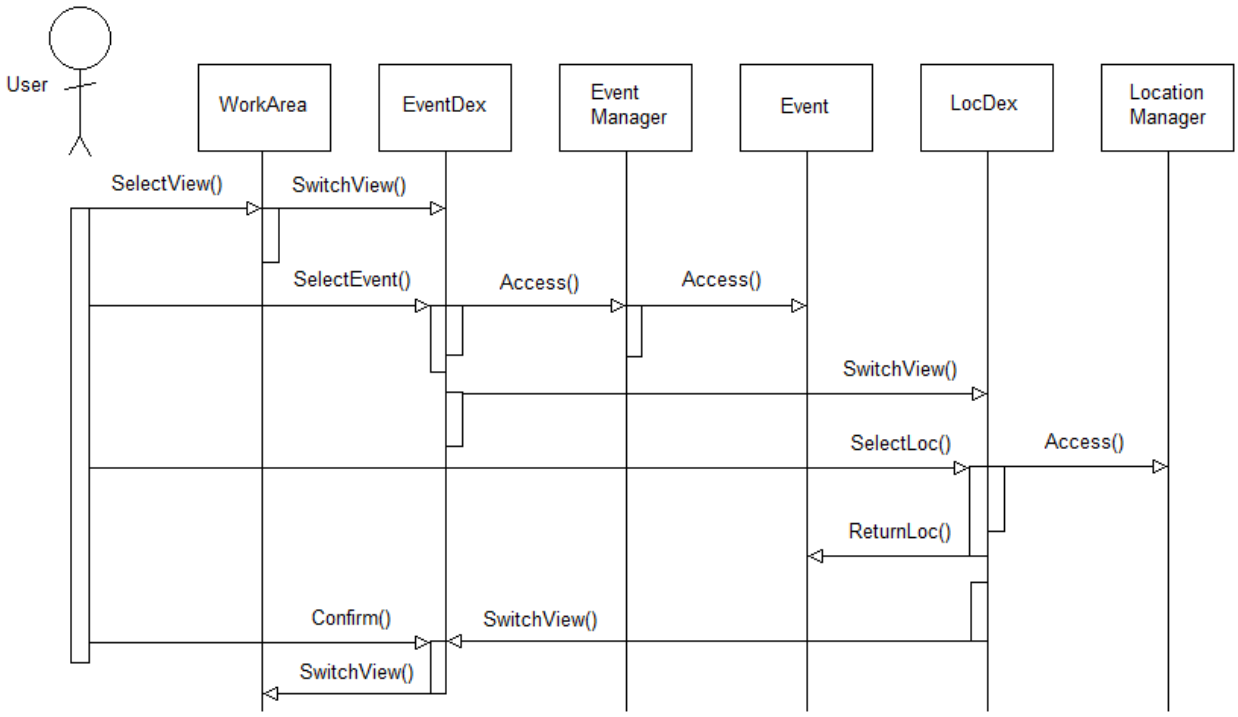


Figure 8: Sequence Diagram for tagEvent

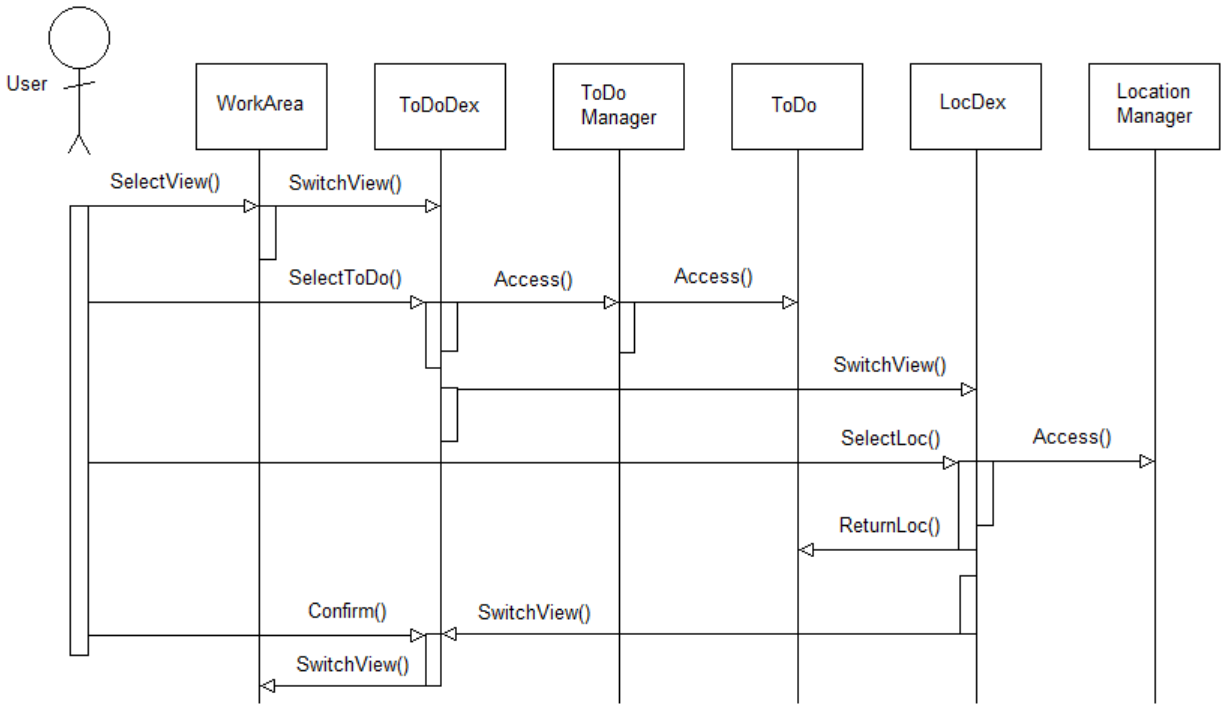


Figure 9: Sequence Diagram for tagToDo

Modeling generalization relationships

Action Class

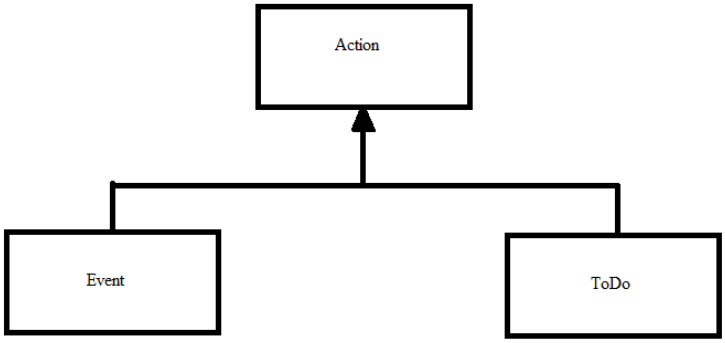


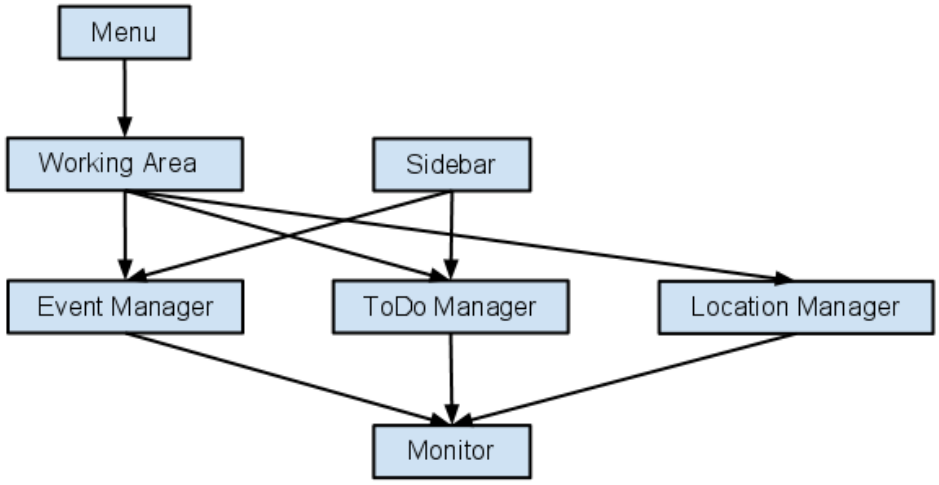
Figure 10

System Design

BerryLife is composed of 7 major subsystems:

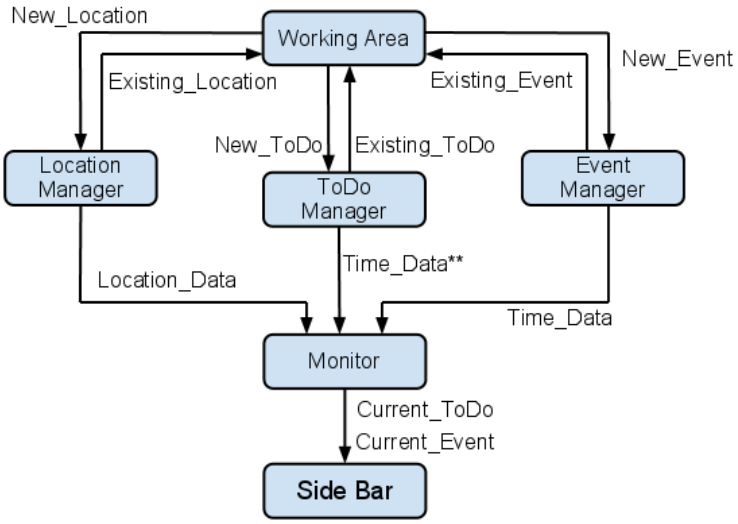
1. Menu
2. Working Area
3. Sidebar
4. Event Manager
5. ToDo Manager
6. Location Manager
7. Monitor

These subsystems are outlined in the following diagram based upon their depth (closest to user interaction at the top).



1.1 - Subsystems by Level

In addition, high-level interactions between the subsystems is outlined in the following diagram using generalized “data”.



1.2 - High Level Data Flow Diagram

Detailed Subsystem Description

The subsystems outlined above will be discussed in further detail; giving information on:

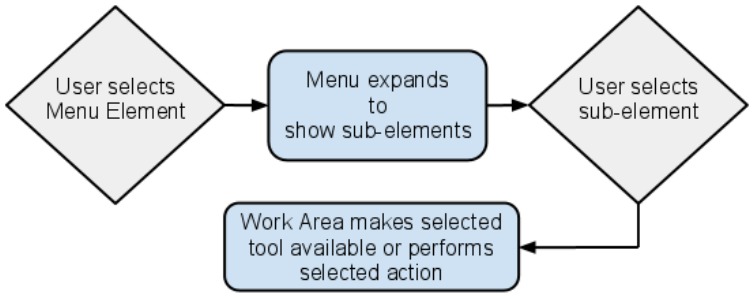
- Subsystem responsibilities
- User interaction with subsystem (where applicable)
- Subsystem classes
- Data flow and interaction with other subsystems

Menu

Responsibilities:

This subsystem is designed to provide the user with an easy, intuitive way to move between the different tools available in the Working Area. It will be composed of four corner buttons, each representing a different family of options, which when clicked on will expand out to make available the relevant sub-options.

User Interaction:



2.1 - Menu User Interaction Diagram

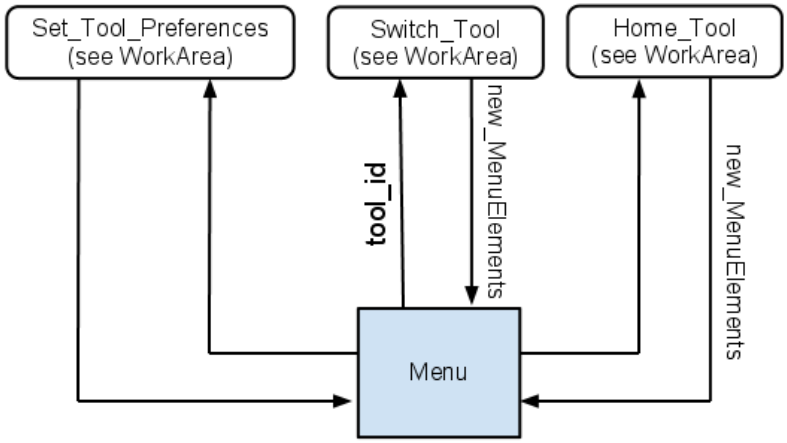
Subsystem Classes:

Menu Class - Contains four MenuElements which will represent the 4 main Menu options. Will also handle user interactions and the graphic elements of the Menu.

MenuElement Class - Contains SubMenuElements which will represent the relevant sub-options. Will indicate to Menu upon user interaction.

SubMenuElement Class - Acts as a link to bring a certain tool to the Work Area or perfrom an action on the tool currently in the Work Area.

Data Flow



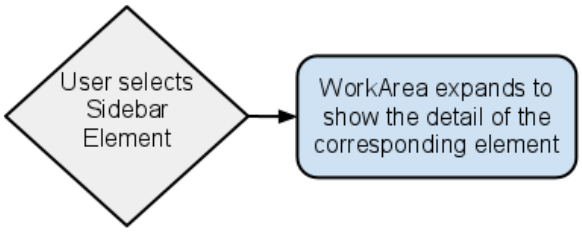
2.2 - Menu Data Flow and Interaction

Sidebar

Responsibilities:

This subsystem is designed to provide user an easier way to view its own list of actions. A vertical bar on the left of the WorkArea will feature dynamic lists of SidebarElements based upon Actions which are currently ‘active’ . User is able to select individual action that is on the sidebar, and views the detail of the corresponding action.

User Interaction:



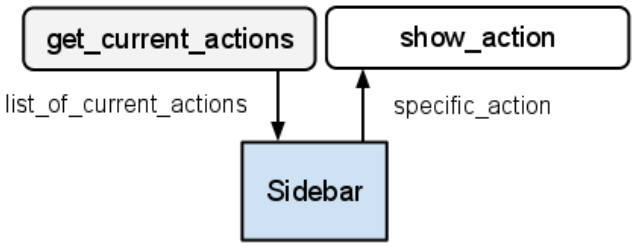
2.3 - Sidebar User Interaction Diagram

Subsystem Classes:

Sidebar Class - Contains SidebarElement that handle user interactions and the graphic elements of the siderbar.

SidebarElement Class - A UI element representing either an Event or ToDo to be displayed by the sidebar. It will be distinguished using a graphic device.

Data flow

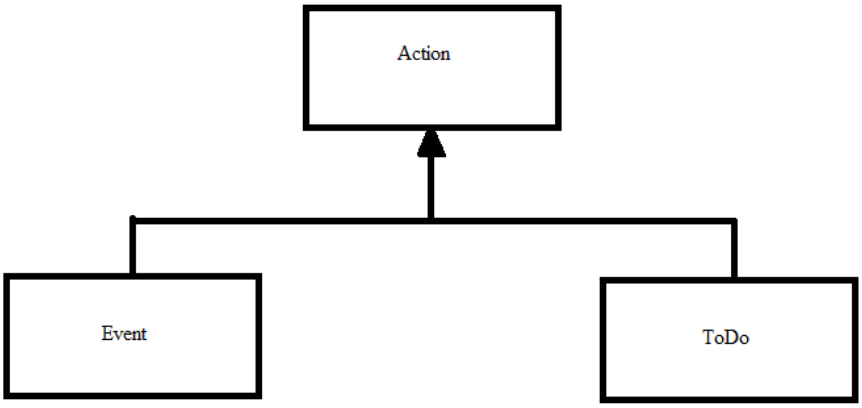


2.4 - Sidebar Data Flow and Interaction

Object Design

Object Description

Action Classes



CLASS Event
RESPONSIBILITY 1 Holds the data of an Event
COLLABORATION 1 Event Manager Class 2 Location Manager Class

CLASS ToDo
RESPONSIBILITY 1 Holds the data of a ToDo list
COLLABORATION 1 ToDo Manager Class 2 Location Manager Class

Class Diagram

EVENT CLASS
eventTime: string NotificationOffset: integer Location: string TimePeriod: string
Public() Event

TODO CLASS
TaskList: string Location: string TimePeriod: string
Public() ToDo

Subsystem Interaction Flow Diagram