

## To-Do:

- assign issues to group
- revise iteration 0 plan

## Implementation

- DSO classes for all  
(planner, vendor, event, eventlist, vendorlist)
- logic needed
- unit tested for complex logic (JUnit)  
AllTests.java
- interface for database
- stub imp of db (List, Map)
- GUI with at least one feature
- same code style
- divide into packages based on architecture

## Architecture

- md file
- sketch
- move incomplete issues to MS 2
- Handin git

# GUI - Add event

EventEase

Welcome Planner!

My Events

Create a new event

EventName

Date

Location

Description

Start

End

- ☐ I'm planning this event for myself
- ☐ I'm planning this event for a client

EventEase

Welcome Planner!

My Events

Event Name	Date
<input data-bbox="280 1634 423 1676" type="button" value="My Vendors"/>	<input data-bbox="451 1634 594 1676" type="button" value="Find Vendors"/>

US : Add event

Class	Event	Event Host
Var:	Event Name	
	Location	
	Date	
	Start Time	
	End Time	
	List Vendors	null
	Description	
Fun :	Constructor	

Edit event

Class: event

Fun :

- setEventName
- setLocation
- setDate
- setStartTime
- setEndTime
- setDescription
- addVendorToEvent (vendor)

Remove Event

class : event

Fun : delete Event

Source Vendors

class : vendor

var : vendorName

ServiceType

description

phoneNumber

email

rate description

rating

fun : Constructor + add to vendorlist

vendorpictures []

Capacity

- find vendors for event (show a list)  
class: vendor list

var: vendor list []  
list title

fun: title by search  
add to list  
remove from list

- filter vendor options by criteria  
class: vendor list

var: filter list []

class: vendor

fun: getServiceType

fun: add filter by service  
remove filter by service  
add filter by cost  
remove filter by cost  
sort Alpha Asc  
sort Alpha Desc  
sort Cost High Low  
sort Cost Low High  
sort Rating High Low  
sort Rating Low High

add filter by Rating  
remove filter by Rating  
reset filters

- view vendor details  
class: vendor

fun: getVendorName  
getVendorDescription  
getVendorService  
getVendorRate  
getVendorRating  
getVendorRateDescription  
getVendorNumber  
getVendorEmail  
getVendorPictures

- vendors respond to queries  
class: vendor dashboard

var: vendor []  
VendorContactName

✱ push to I2?  
US: Accept/Reject  
Event Task requests  
↑ Dupe?

class: vendor Requests

var: vendor status  $\rightarrow$  bool

planner request	(Pending)
vendor request	(Action required)
vendor accept	(Accepted)
vendor declined	(Declined)
awaiting Event	(Wait for Event)
order Progress	(order in Progress)
Order Ready	(order Ready)
Order Complete	

fun: set Status

Constructor  $\rightarrow$  when planner sends request

# Sort Events

- event sorting

class : event list

Var : events[]  
Num events

fun : constructor → when planner makes account

sortBy Event Date

sortBy Num Vendors Asc

sortBy Num Vendors Desc

sortBy Num Guest Asc

sortBy Num Guest Desc

add Event To List

remove Event From List



Architecture