

Walkthrough order:

overview schema + log in: Emily

calendar (explain events & time blocks, make event to start in 15 minutes, edit an event/time block, delete an event/time block): Billy & Ivan

tasks (show on context view & all tasks view, add/edit/delete): Mark

data (show on category view, go to one of the data views, change the category, open that category view, explain how you can add your calendar from ics, contacts from vCard, rest from phone): Kathryn

phone app (show everything, including settings, & end on notification): Eric

reminders (explain popup reminders vs notifications vs email reminders, show how to set default times on settings panel, show email on demo gmail account): Elfalem

admin panel --> config file --> schema, show tables, show the user activity log: Team 10

security (describe how awesome we are): Josh

by Thursday:

in SVN: code, config, apk, security logs, ssl cert, open source license

Josh: calendar code, SVN, put security logs in SVN

Elfalem: sanitize inputs, repeating events, set up cron jobs

Eric: send data via json, email out APK & put in SVN

team 10: config file, data from json to database, email check-in & follow-ups, get ssl cert, repair delete account

Mark: try to fix css

Billy & Ivan: fix ALL THE THINGS

Kathryn: vCard parser, log, end survey

later:

in SVN: eng logs, meeting minutes, WATIR tests, all validation stuff

everyone: validation testing, cost analysis

24 November:**things we may not have time for:**

- email service
- dependent tasks
- recurring tasks (events done, not tasks)
- reminder via text message
- ~~postponing context blocks~~ (completed on context page. A click postpones a block for 15 minutes)
- weekly check-in email: list of upcoming tasks/events/time blocks
- weekly follow-up email: overdue tasks, tell to add more time blocks
- buffer time: when a user opens a context view, a script will compare the total remaining time estimates of tasks (+20%) to the total time blocks for that context and ask the user to set more time blocks if the task time is greater than the time blocks

stuff no one has claimed, but we need someone to do:

- help (FAQ) files (Kathryn)
- [SSL certificate](#)
- [open source license \(put a license.txt in Bootstrap-Billy\)](#)
- validation surveys (Kathryn)
- template for log validation testers will use to record interactions over the day (Kathryn)

Team 3:

Eric:

- Per task reminders
- ~~Event and context block layout, display on calendar/category view~~
- ~~Repeating event/blocks~~
- ~~Auto-select view to jump to on app launch~~
- ~~Get userid on login~~
- ~~Pull task/event data from server~~ (done, need user acct with valid data for testing)
- ~~Finish sync service~~
- ~~get contacts/calls/texts~~, send to server (info gathering done, need php script to send to)
- ~~display contacts/calls/texts in app~~

Emily:

- read & comment on validation narrative
- test reminders, index.php, tasks, calendar, & contacts upload
- tag data
- sanitize all selects
- settings.php: ~~delete account, submit updated information to database~~
- navbar: ~~add new tasks to database~~
- ~~edit/delete current category/context~~
- ~~individual cat/con active li & change color~~
- ~~sanitize inputs~~
- ~~index.php: check for current event/time block, open this page (calendar otherwise)~~
- ~~reminders: query every 1 minute, popup, ok/ignore/postpone~~
- ~~update_session_vars.php98~~
- ~~write_add_from_json.php (sanitize inputs)~~
- ~~category/context views: display all data~~
- ~~data views: display all data~~
- after Monday: add admin panel link to navbar, settings updates, link to APK, make sure all inputs are sanitized & using non-root account, delete old code

Elfaalem:

- maintain DB [up to date as of now]
- help Emily with frontend web stuff

Josh

- set up SVN for commit
- create new user for Apache & transfer DB set up

Paarth

- disabled-users
 - ⊖ admins can deactivate users
 - ⊖ deactivated users cannot login
- Features list
- Validation Exercises

Team 9:

Kathryn:

- finish up contact stuff
- put it in /var/www/ on vulgarity
- indicate which is the main file
- ensure it works again
- make sure it is integrated into the database (secure_login on vulgarity)
- help (FAQ) files
- validation surveys
- template for log validation testers will use to record interactions over the day

Billy & Ivan:

- ~~Move calendars and update GUIs~~
- ~~Write script for getting color based on category~~
- ~~Fill form when updating~~
- ~~Push events to database~~
- ~~Push contexts to database~~
- ~~Pull events from database with color~~
- ~~Pull context from database with color~~
- ~~Pull contexts with filters~~
- ~~Update events/contexts~~
- ~~pull Repeating Events with color pull Repeating Context with color~~
- ~~Add import ics button to calendar.php~~
- ~~add allDay to timeBlock~~

Team 10:

~~Forgot Password - Email user w/ password~~

~~Change Password~~

~~Unique E-mail~~

~~Config Files~~

~~Remember Me~~

~~New User - Add 4 Context entries with user_id to table~~

~~User activity log~~

17 November:

To do list for EVERYONE (add/cross off things as necessary):

Team 3:

Elfalem:

~~ajax.php/function.php: get working for web & android access to database given a session~~

ID

Eric:

~~dependent tasks~~

~~task storage~~

~~background notification & sync service~~

~~send calls + texts + contacts~~

~~send final APK to Emily (to put on website)~~

Emily:

~~see list on vulgarity.cs.umd.edu/Bootstrap/terp-tasker-bootstrap-todo.txt (it's kind of long)~~

Josh:

~~write security tests (if needed)~~

~~launch Nessus and Nmap against the server~~

~~investigate remaining open source tools~~

Paarth:

~~validation test narrative~~

Team 9:

Billy & Ivan:

~~send Josh link to open source calendar parser~~

~~load from user ID~~

~~make sure events match database schema (see google drawing)~~

~~event reminders~~

~~move to vulgarity~~

Mark:

~~send Josh link to open source tasker~~

~~dependent tasks~~

~~load from user ID~~

~~task reminders~~

~~put "task title" in search bar~~

~~move to vulgarity~~

Kathryn:

~~find & implement vcard parser~~

~~help (FAQ) files~~

Team 10:

~~set up database~~

test ~~add/edit/delete~~

write config file

make sure input validation, remember me, forgot password, ~~recaptcha work, user activity~~
table

write & run WATIR tests

What Josh has done so far and what he has left to do:

Josh has updated the server quite a bit, checked the security of all the open source utilities we are using besides the calendar parser and task list. Need to do those, but don't have the open source tools for them.

TODO:

This weekend - launch Nessus and Nmap against the server to figure out if there are any major security holes (this might leak into next week and weekend because the VMs are being dumb and I didn't set checkpoints when I should have)

Next weekend - have all of the open source tools investigated

THINGS JOSH NEEDS DURING THE MEETING:

Open source tasker we're using (if we're not writing it ourselves)

Open source calendar parser we're using (if we're not writing it ourselves)

Predicted Timelines:

Emily: 14 Nov: home page, settings page, navbar, move all files to PHP, header

19 Nov: data views, context/category views

21 Nov: reminders, validation narrative written

Elfalem (& Paarth): 15 Nov: finalize whole database schema, and draft data object specifications

Guys please put your timelines here. I can't make it to class today to bother all of you so I'll just have to do it over the Internet - Emily (8 Nov)

I give up - Emily (14 Nov)

Links:

Calendar:

<http://vale.cs.umd.edu/fullcalendar/demos/calendar.html>

Todolist:

<http://vale.cs.umd.edu/myTinyTodo/mytinytodo/index.php>

Bootstrap:

<http://vulgarity.cs.umd.edu/Bootstrap>

November 1-5: Team & Individual Work

Josh's security upgrades - make sure your stuff isn't broken (it shouldn't be)

Updated the following on the server:

apt apt-transport-https apt-utils dpkg initramfs-tools initramfs-tools-bin

libapt-inst1.4 libapt-pkg4.12 libglib2.0-0 liblockfile-bin liblockfile1

owncloud procs python-lazr.restfulclient tzdata wpasupplicant

Previous version of PHP: 5.3.10

Vulnerabilities that made us upgrade

(http://www.cvedetails.com/vulnerability-list/vendor_id-74/product_id-128/version_id-125887/PHP-PHP-5.3.10.html)

Added repository listed here:

(<http://askubuntu.com/questions/109404/how-do-i-install-latest-php-in-supported-ubuntu-versions-like-5-4-x-in-ubuntu-1>)

Upgraded apache and php:

The following NEW packages will be installed:

apache2-bin apache2-data build-essential debhelper dh-apparmor dpkg-dev fakeroot g++
g++-4.6 gettext html2text intltool-debian libalgorithm-diff-perl libalgorithm-diff-xs-perl
libalgorithm-merge-perl libdpkg-perl libgd3 libgettextpo0 libjson-c2 liblua5.1-0
libmail-sendmail-perl libonig2 libqdbm14 libstdc++6-4.6-dev libsys-hostname-long-perl
libunistring0 libvpx1 make php5-json php5-readline pkg-php-tools po-debconf

The following packages will be upgraded:

apache2 apache2-mpm-prefork apache2.2-bin libapache2-mod-php5 php-pear php5 php5-cli
php5-common php5-dev php5-gd php5-mysql php5-sqlite
12 upgraded, 32 newly installed, 1 to remove and 0 not upgraded.

Current version of PHP: 5.5.5-1

Apache upgraded as well.

Explanation of .php files (for team 10)

support.php & index.php currently do nothing

index.php is an example of how to use the generatePage function

support.php is based on functions.php in the Login directory

it takes a string (the body, which goes in span9) and generates the page around it, including the header and sidebar

the list of categories & contexts need to be made from a query

here is the code block that is supposed to do that:

```
$result = $mysqli->query("SELECT * FROM Categories"); //get list of categories
$num = $result->num_rows;
$i=0;
while ($i < $num) { //for each category
    $name=mysqli_result($result,$i,"name"); //get name of category
```

```

$views = <<<EOPAGE
<li><a href="$name.html">$name</a></li> //this link will eventually be a get request (eg
category.html?cat="$name")
EOPAGE; //append category link to navbar
echo $views;
$i++;
}
$result->close(); //close query

```

Emily's stuff for later:

nav bar: add tasks from sidebar

settings: buffer time, weekly emails

move all to php

change links to gets

set active li

home page: explain Terp Tasker, link to APK

calendar: combine with team 9

upload button

category/context views: query from get, display all data

add/edit/del tasks & events

context view only: get warning for more time

edit/delete current category/context

data views: add/edit/del tasks

tag stuff

upload button or instructions

reminders: query every 1 minute, popup, ok/ignore/postpone

other: add logo

make sure remember me & forgot password work

make sure settings submit & cancel buttons work

database stuff for website:

home page: ~~make logins & registration work (combine with login scripts)~~

on login: start session, check for current event/time block, open this page (calendar otherwise)

header: set "Terp Tasker" link to same as above

header: get name

make logout link work

navbar: query context & category names, create links in sidebar

when new category/context created, add to database

settings: populate name & email (placeholder="") and reminders (change <option> to <option selected="selected">) from

submit updated information to database

October 31: Meeting in Class

team 3:

currently auditing open source code & database security

proposal is 2 days from submission

Twitter bootstrap front end by this weekend

have a rough Android app that works on Gingerbread+

team 9:

having trouble with recurring events

may need to add a new instance for each repeat in database

need to add start date+time, end time, & reminders to tasks

need to add reminders to events

later: contacts

team 10:

accounts basically done

register now at vulgarity.cs.umd.edu

working on IMAP & integration with teams 3 & 9

later: add control panel for admin account on roll-your-own instance

cross-team:

Eric & Matt will be working on Android app, specifically accounts

Billy & Eyobe will be working on getting a calendar saved under an account in the database

Later:

validation test stuff: configuration file, FAQ/help files, narrative, surveys

October 28th: Team 10 Meeting

Completed tasks:

Login

Registration

Database setups

Security (? Josh take a look)

Login and registration are used from code found at:

<http://www.wikihow.com/Create-a-Secure-Login-Script-in-PHP-and-MySQL>

No api or anything, straight up php, html, mysql, and javascript

Didn't really do anything in terms of aesthetics, or actions after logging in as we don't have anything concrete for design.

For this upcoming week, we'll probably implement more features for the registration script, such as making sure a user can't register with a pre-existing email address (should be easy).

Need more information from the rest of the team to decide on what to continue on.

October 27th: Team 9 Meeting

General things:

For the calendar, we will be using this:

<http://vale.cs.umd.edu/fullcalendar/demos/calendar.html> ← this link has changed from when the meeting minutes were previously posted (updated 10/29)

which we got from this site:

<http://arshaw.com/fullcalendar/>

right now it's hard coded to take data from a specific calendar and data doesn't get saved at all

For the todo list/tasking, we will be using this:

<http://vale.cs.umd.edu/myTinyTodo/mytinytodo/index.php>

which we got from this site:

<http://www.mytinytodo.net/>

Here data does persist as it gets saved into a mysql database that for right now the tool itself creates.

We also defined data that a **task** needs to store:

- Due Date
- Context
- Description
- Category
- Time Estimate
- Associated Files (Contacts, e-mails, files, etc.)

And data that an **event** needs to store:

- {Fields from .ics file}
- Category
- Associated Files

Any times should be stored in UTC time in the database (GMT +0) in order to be consistent, and then the front end can take care of converting it to whatever the user needs

Over the next week, here's what we'll be working on:

Mark - Looking into mytinytodo more to learn the code and perhaps document it (it is sparsely documented right now and doesn't come with an api), looking into changing where it writes the data to, and looking into generating to-do lists based on context

Billy - Interface for adding events to the calendar

Kathryn - Looking into integration (both front and backend) between the two open sourced programs, modifying the calendar code to push data into the database (will need to talk to Team 10 some), and once Ivan creates a button, trying to upload her own .ics calendar

Ivan - Get calendar to pull information from database (will need to talk to team 10 some), and create and "upload .ics calendar" button

October 27th: Team 3 Meeting

android:

Eric has a demo interface shell running in 4.0+

- display: calendar (is a 4.0+ API call), context/category
for now: pull from ical file on vulgarity; display fake contexts & categories
- pull stuff from & add stuff to stuff database
add sync option to side bar
sync every 15 minutes
send contacts, texts, & call history to website
- app notifications

security:

Josh will be auditing all open source code & ensuring the account system is secure.

- need open source code list for audit
list of source & function calls
looking for sanitizing inputs

OPEN SOURCE CODE LIST

- IMAP
- calendar parser
- calendar display
- task list
- Twitter bootstrap
- Android libraries

web shell:

Emily will set up a framework in Twitter Bootstrap & working with Paarth + team 9 to get data to display.

integration:

Paarth & Elfalem will be setting up the back end framework with team 10.

By Thursday, team 3 will have finished:

- Josh will audit all open source code.

- Paarth & Elfalem will have initial database mapping, email all subteams asking for how their data is formatted.
- Emily will have the web shell (menu sidebar, empty divs) finished.
- Eric will have fake categories & contexts, basic login stuff done.

October 13th

Proposal needs editing, specifically “competitors section”

- Either do a Terp Tasker vs Competitor style
- Or do a ranking system

E-MAIL API: `php.net/manual/en/function.imap-headerinfo.php`

- squirrelmail
- Addition to php files

JQUERY CALENDAR API: <https://github.com/robmonie/jquery-week-calendar/wiki>

- Open-source from github
- Right-click to add events

October 12th

Goals for this meeting:

- make decisions about APIs, open source tools & priorities
- walk through set up & scenarios (ie draw a mockup on the white board and act out user cases)
- divy up proposal writing
- pick next people & time to meet with Purtilo

October 10th

Features Review:

gtd: tagging & blocking time

emails: pull from header

calendar: instructions for downloading

make our version downloadable

no syncing

look into ical <-> icloud sync & outlook <-> outlook.com sync?

contacts/calls/texts: from phone

from owncloud

files:

drive, owncloud, dropbox

do we need files at all?

read only + links?

learning: intro survey

notifications

3 locked-in profiles

buffer time

leak time

change difficulty & estimated time to easy/medium/hard to make task entry easier for the user
accommodating pre-meeting prep time or travel time:

allow user to set how early reminder alarm goes off (suggest adding time for these things)

to do:

find & download open source packages for calendar parser, to do list, & IMAP to VM (vulgarity)

Calendar Parser:

...

Open Source Taskers:

<http://sourceforge.net/directory/business-enterprise/todo-lists/os:linux/freshness:recently-updated/>

<http://www.mytinytodo.net/>

<http://www.plancake.com/open-source>

<http://www.taskfreak.com/original>

<http://getontracks.org/>

Email Service:

IMAP