

15-437
Final Project Specification

Team: Winter is Coming

Members: Gal Fisher, Aurelio Puebla Gutiérrez, Laura A. Miller

Date: 11-13-2016

Document Overview

This document will contain detailed specifications on the CaPS project. To recap the purpose of the project:

The driving idea behind this project is to create a webapp that will lower the “activation energy” for students to get psychological help from a counselor at CAPS and aid in a more comprehensive treatment program. Often, even making a phone call is a large barrier to those with mental health issues, so our app would make it easier for them to get an appointment or get emergency help by using a chat session rather than calling. Counselors will and students can create profiles and be matched with a matching algorithm for a best fit for treatment. The app will also implement a gamified “homework” system that will allow the counselor to assign various exercises to the student so both can keep track of progress between sessions. The exercises will be able to be used for both diagnosis and treatment.

These features will make it easier to have a long-term relationship with CAPS or get support for a short-term crisis. Students can put in as much or as little effort as they want in their treatment.

We will provide specifications for the user interface and site design through wireframes and textual descriptions, the specifications for the gamified “homework” system through the task framework, and the specifications for the CaPS backend via Django models and views.

Wireframes

First, we will demonstrate the design and intended functionality of our web application through wireframes. They are divided into three subsections:

1. Main pages
 - Main Page, Services, Resources, Contact, Registration Welcome/Information
2. User pages
 - Registration, Student Profile (private and public), Counselor Profile (private and public),
3. Assignment pages
 - Assignments View for Students, Assignments View for Counselors, Assignment Statistics, Assignment Create/Edit

Main Page

Carnegie Mellon University[Login](#)

Counseling and Psychological Services (CaPS)

[Get Support](#)[Services](#)[Resources](#)[Contact](#)

We're here to help.

[CaPS](#) provides a safe, confidential environment for students to talk about personal and academic concerns. We also provide consultation services for faculty, staff, and family members.

[Call, stop by,](#) or chat with a scheduler below to schedule a consultation.

About Us

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio euismod sit amet. Quisque laoreet sem sit amet orci ullamcorper at ultricies metus viverra. Pellentesque arcu mauris, malesuada quis ornare accumsan, blandit sed diam.

Daily Health Tip

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia

[Get Emergency Help](#)[Set up an Appointment](#)

The main page is the main point of access for the site. It will contain a high-level description of CaPS and buttons to get support, see what services CaPS offers, find links to external resources, and find standard contact information. Since “Services”, “Resources”, and “Contact” are all static pages with obvious content, so they will not have wireframes dedicated to them. The bar at the bottom is a floating bar that will stay in place as the user scrolls and contains chat buttons. Each button and link on all pages (not just the home page) will have mouseover dialogues describing what they are. The “Daily Health Tip” box is a stretch goal.

Main Chat (Part of Main Page)

Carnegie Mellon University[Login](#)

Counseling and Psychological Services (CaPS)

Get Support

Services

Resources



Contact

You are chatting with Elaina

You are chatting with a trained professional. Everything you say will be kept confidential and your name will be kept anonymous unless there is an immediate threat to yourself or others.

Elaina: Hello, what is your emergency?

Type your message here...



Set up an Appointment

A view of the chat box once the “Get Emergency Help” button is clicked. It will contain the name of the emergency contact the person is speaking to, and a short dialogue explaining anonymity. The “Set up an Appointment” button will result in a similar chat box, but instead of chatting with an emergency responder, they will be chatting with a secretary/scheduler at the front desk. A stretch goal is to include video and audio contact in the chat.

Registration Welcome/Info

Carnegie Mellon University[Login](#)

Counseling and Psychological Services (CaPS)

[Get Support](#)[Services](#)[Resources](#)[Contact](#)

We're glad to help.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio euismod sit amet. Quisque laoreet sem sit amet orci ullamcorper at ultricies metus viverra. Pellentesque arcu mauris, malesuada quis ornare accumsan, blandit sed diam.

Register

[Get Emergency Help](#)[Set up an Appointment](#)

Once the “Get Support” button is clicked, it will turn red to show it is currently selected. This will take the user to a page that welcomes them and describes the registration process and the purpose of making an account. The “Register” button will lead the user to the registration page.

Registration

Carnegie Mellon University[Login](#)

Counseling and Psychological Services (CaPS)

[Get Support](#)[Services](#)[Resources](#)[Contact](#)

You're on your way.

Everything will be kept confidential until you are matched with a therapist.
Then only they will have access to your information, and you can revoke access at any time.

Tell us about yourself

First Name:	<input type="text" value="First Name"/>	Preferred Name:	<input type="text" value="Preferred Name"/>
Last Name:	<input type="text" value="Last Name"/>	Age:	<input type="text" value="Age"/>
Major:	<input type="text" value="Major"/>	Gender:	<input type="text" value="Gender"/>

Reasons for seeking help:

<input checked="" type="checkbox"/> Academic Stress	<input checked="" type="checkbox"/> Another reason	<input checked="" type="checkbox"/> Another reason
<input checked="" type="checkbox"/> General Stress	<input checked="" type="checkbox"/> Another reason	<input checked="" type="checkbox"/> Another reason
<input checked="" type="checkbox"/> Anxiety	<input checked="" type="checkbox"/> Another reason	<input checked="" type="checkbox"/> Other (Please specify)
<input checked="" type="checkbox"/> Depression	<input checked="" type="checkbox"/> Another reason	<div><input type="text" value="Other reasons..."/></div>

Register

The register page will reinforce the promise of anonymity to the user to bring down the barrier to filling out the relevant information. This form gets basic user information (and andrewID and a password, although it is not shown here for authentication) and may be extended later to include more.

Student Profile

Carnegie Mellon University

[Home](#)

Profile ▼

Counseling and Psychological Services (CaPS)



[Change Profile Picture](#)

First Name (Preferred Name) Last Name

Age / Gender

Major

Counselor: [Counselor Name](#)

[Other Counselors](#)

Next Appointment: mm/dd/yyyy

[Reschedule](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque

Reasons for seeking help:

☒ Academic Stress

☒ Another reason

☒ Another reason

☒ General Stress

☒ Another reason

☒ Another reason

☒ Anxiety

☒ Another reason

☒ Other (Please specify)

☒ Depression

☒ Another reason

Other reasons...

Exercises

▼ Name	▼ Status
Exercies 3	Not Started
Exercise 2	In Progress
Exercise 1	Finished
Compatibility Measures	Recurring

[Get Emergency Help](#)[Set up an Appointment](#)

The student profile page will contain an “ID Card” at the top containing basic information about the student and a picture. Below is a bio created by the student to describe themselves in whatever amount of detail they wish. The “Reasons for seeking help” section may be changed at any time by checking or unchecking checkboxes to allow the student to update their personal situation. Below that is a section for exercises where the student can see their assigned exercises and what their completion statuses are. The exercises will be able to be clicked on to bring them to the assignment page for that exercise. The first exercise, “Compatibility Measures,” will be automatically assigned to help to match the student to a counselor. A student may still open up a chat window for emergency help or to set up an appointment if they wish.

The public view of the student’s profile will only be able to be seen by their counselor. The dropdown box on the top right will contain links to logout, edit the student’s profile, or deactivate the account if the student thinks they are no longer in need of treatment. Accounts can also be deactivated via the Django moderator page if the student graduates or otherwise leaves the school.

Counselor Profile

Carnegie Mellon University

[Home](#)

Profile ▲

Counseling and Psychological Services (CaPS)

Edit

Logout



[Change Profile Picture](#)

First Name (Preferred Name) Last Name

Age / Gender

Degree obtained / School

Position Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque

Students

▼ Student Name	▼ Exercise	▼ Status
Student 1	Exercise 1	Not Started
	Compatibility Measures	Recurring
Student 2	Exercise 2	In Progress
	Exercise 1	Finished
	Compatibility Measures	Recurring

Schedule

◀ April 22, 2012 ▶						
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21

The counselor's profile is the same as the student's in most ways. The only difference is the "ID Card" will contain different information, there are no reasons for seeking help, the "Assignments" section is replaced with the "Students" section, and there is a schedule of availability. The counselor's profile will be able to be viewed by any student.

The student table will show all of the counselor's students and what assignments have been given to them. A student can be clicked on to go to the "Assignment View for Counselors" page and an assignment can be clicked on to go to the "Assignment Statistic" page for that assignment. If a student graduates or deactivates their account, they will no longer be shown in the table.

The schedule is a stretch goal that will show when the counselor has appointments and when there are availabilities and can be edited directly. In the public view, if the schedule is full, there will be some text saying that the counselor is not accepting new patients. If the schedule is not full, it will show availabilities to any student that looks at the page.

Assignments View for Students

Carnegie Mellon University

[Home](#)

Profile



Counseling and Psychological Services (CaPS)

Assignment X



[Get Emergency Help](#)

[Set up an Appointment](#)

This page shows the assignment that a student is working on in a view screen. The assignment can be anything that fits to the framework, so the window can contain a variety of different things. It also gives a progress bar below it to indicate how far along a student is in the assignment.

Assignments View for Counselors

Carnegie Mellon University

[Home](#)

Profile



Counseling and Psychological Services (CaPS)

Exercises for Student X

▼ Exercise	▼ Status
<input checked="" type="checkbox"/> Exercise 1	Not Started
<input type="checkbox"/> Exercise 2	Not Assigned
<input type="checkbox"/> Exercise 3	Not Assigned
<input checked="" type="checkbox"/> Compatibility Measures	Recurring

This page shows the exercises currently assigned to each student. The counselor has the power to assign exercises to students by checking or unchecking the checkboxes on the left and can see the status of the assignment on the right.

Assignment Statistics

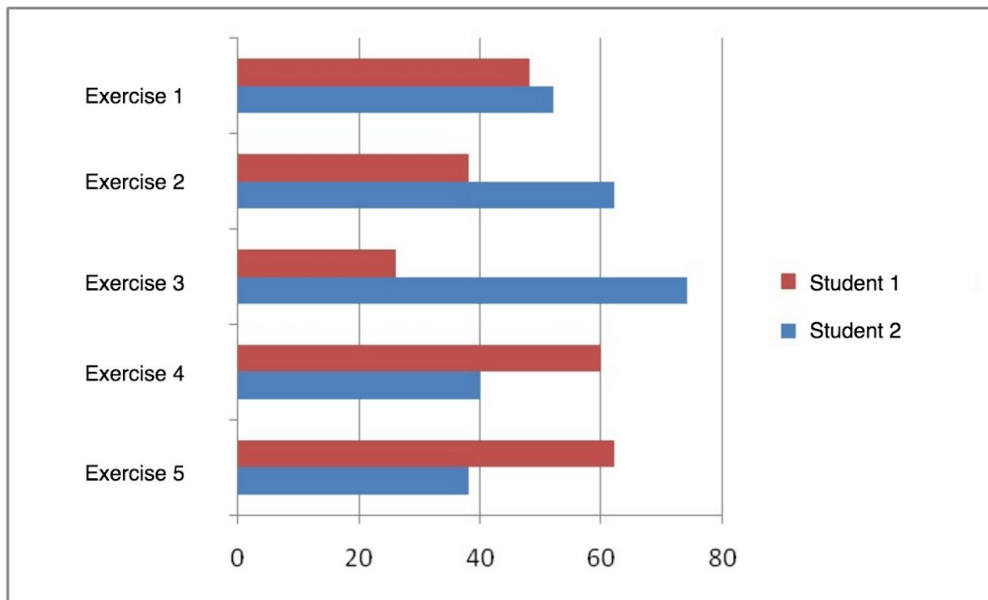
Carnegie Mellon University

[Home](#)

Profile ▼

Counseling and Psychological Services (CaPS)

Assignment Statistics



Filter Results

Exercise

Start Date

End Date

Exercises

Student

☒ Student 1

☒ Student 2

☐ Student 3

This page is viewable both by students and counselors and provides filtering options to get more useful feedback, except that only the counselors will be able to filter by student (each student can only see their own exercise statistics). Various things will be viewable such as time taken to complete an assignment or a question on an assignment, mean, variance, etc.

Assignment Create/Edit

Carnegie Mellon University

[Home](#)

Profile ▼

Counseling and Psychological Services (CaPS)

Create and Edit Exercises

Create New	Edit Existing
<div>Exercise 1</div> <div>Exercise 2</div> <div>Exercise 3</div> <div>Exercise 4</div>	

This page allows counselors to create new assignments based off of existing templates or edit existing assignments. Assignments will originally be added by a superuser or administrator, and counselors will be able to extend them to their needs from there. The “Compatibility Measures” assignment will not be able to be edited. Since each exercise is based off of the framework, they will show up on the right side of the box to be edited and their layout is not yet determined.

Task Framework

The task framework is an abstraction for exercise assignment and response management. It's designed so that it can be integrated in many projects that need such functionality such as online quizzes, games, surveys, and so on. It's made so that it'd be easy to keep track of assignments, all the user responses and interactions with the task, statistics about the performance, and status/history of the assignments. It'll be used in this project as a way of managing the matching quiz, diagnosis tests, and the exercises for the students in CaPS.

Models:

```
from django.db import models
```

```
class Option(models.Model):
    number = models.IntegerField() # Id for the given question
    text = models.CharField(max_length=1023)
    correctness = models.BooleanField()
    question = models.ForeignKey(Question, related_name='options')
```

```
class Question(models.Model):
    number = models.IntegerField() # Id for the given task
    text = models.CharField(max_length=1023)
    # options (defined in: Option)
    time_limit = models.DurationField(blank=True)
    task = models.ForeignKey(Task, related_name='questions')
    # Note: Validate options id to be 1-N
```

```
class Task(models.Model):
    name = models.CharField(max_length=255, unique=True)
    interface = models.FileField()
    # questions (defined in: Question)
    # Note: Validate questions id to be 1-N
```

```
class InteractionUnit(models.Model):
    task_interaction = models.ForeignKey(TaskInteraction, related_name='interaction_units')
    question = models.ForeignKey(Question)
    selected_options = models.ManyToManyField(Option)
    correctness = models.BooleanField(default=False)
    start_datetime = models.DateTimeField(auto_now_add=True)
    end_datetime = models.DateTimeField(auto_now=True)
    completed = models.BooleanField(default=False)
```

```
class TaskInteraction(models.Model):
```

```

assigned_task = models.ForeignKey(AssignedTask, related_name='task_interactions')
# interaction_units (defined in: InteractionUnit)
start_datetime = models.DateTimeField(auto_now_add=True)
end_datetime = models.DateTimeField(auto_now=True)
completed = models.BooleanField(default=False)

```

```

class AssignedTask(models.Model):
    task = models.ForeignKey(Task)
    # task_interactions (defined in: TaskInteraction)
    tasks_profile = models.ForeignKey(TasksProfile, related_name='assigned_tasks')
    repetitions = models.IntegerField(blank=True)
    assignation_datetime = models.DateTimeField(auto_now_add=True)
    deadline_datetime = models.DateTimeField(blank=True)

```

```

class TasksProfile(models.Model):
    user = models.ForeignKey(User)
    active_assigned_task = models.OneToOneField(AssignedTask)
    # assigned_tasks (defined in: AssignedTask)

```

View Actions:

```

def get_statistics(tasks_profiles, tasks):
    """ Gets basic statistics for the profiles provided at the specified
        tasks. """
    raise Http404("Not implemented")

def start_task(task_profile, task_name):
    """ Starts a task for the user. """
    raise Http404("Not implemented")

def get_question(task_profile):
    """ Gets the question with its options and time-constraints. """
    raise Http404("Not implemented")

def send_answer(task_profile, options):
    """ Receives the answer from the user and updates the interaction. """
    raise Http404("Not implemented")

def end_task(task_profile):
    """ Ends the current task for the user. """
    raise Http404("Not implemented")

# Abstraction of complex queries to the db of the framework could be added

```

CaPS Backend

The CaPS Application is the main portion of the site. It is composed of various models and actions to represent profiles for and serve content to students and counselors. It will keep track of information regarding identifying descriptors, biographies, meetings, as well as student or counselor specific information.

Models:

```
class Student(models.Model):
    andrew_id = models.CharField(max_length=30)
    password = models.CharField(max_length=50)
    first_name = models.CharField(max_length=30, default="", blank=True) # blank=True -
student can remain anonymous
    last_name = models.CharField(max_length=30, default="", blank=True)
    preferred_name = models.CharField(max_length=30, default="", blank=True)
    age = models.IntegerField(null=True, blank=True)
    MALE = 'M'
    FEMALE = 'F'
    GENDER_CHOICES = ( (MALE, 'Male'), (FEMALE, 'Female') )
    gender = models.CharField(max_length=1, choices=GENDER_CHOICE)
    major = models.CharField(max_length=15, default="", blank=True) # CS, ECE, ...
    seeking_help_reasons = models.CharField(max_length=15, default="", blank=True) #
implement using choices ?
    year = models.CharField(default="", blank=True) # Freshman, Sophomore, ... ,
Graduate, Other
    counselor = ForeignKey(Counselor, on_delete=models.CASCADE,
related_name="students")
    is_active = models.BooleanField(default=True) # True if the user is getting help from
CaPS
    # meetings (defined in Meeting)

class Counselor(models.Model):
    email = models.EmailField(max_length=50)
    password = models.CharField(max_length=50)
    first_name = models.CharField(max_length=30, default="")
    last_name = models.CharField(max_length=30, default="")
    position_title = models.CharField(max_length=30, default="Advisor")
    age = models.IntegerField(null=True) # no blank=True - we want the Counselor to fill all
the details
    MALE = 'M'
    FEMALE = 'F'
    GENDER_CHOICES = ( (MALE, 'M'), (FEMALE, 'F') )
```

```

gender = models.CharField(max_length=2, choices=GENDER_CHOICE)
seniority = models.IntegerField(null=True)
degree = models.CharField(max_length=50, default="", blank=True)
school = models.CharField(max_length=50, default="", blank=True)
about = models.TextField(max_length=1023, default="", blank=True)
Picture = models.ImageField(upload_to="")
# meetings (defined in Meeting)

```

```

Class Meeting(models.Model):
    student = models.ForeignKey(Student, related_name="meetings")
    counselor = models.ForeignKey(Counselor, related_name="meetings")
    start_date_time = models.DateTimeField(blank=False) # field will be required in the form
    end_date_time = models.DateTimeField(blank=False)

```

View Actions:

```
## Student actions
```

```

def home(request)
    # the landing page - first page that is shown when entering the site
    # will providing information about caps and a log in/registration button
    # if the user is already logged in he can set up appointment or access his profile page
    # in case of an emergency - a student can
    return HttpRequest("Not Implemented")

def chat(request, emergency):
    # in the front page of the website, there will be an emergency button that will allow a
    student, even if not logged in, to talk to a CaPS counselor immediately, if he's in distress.
    # in addition, if emergency=False, there will be also a chat, but this time only in working
    hours and with CaPS secretary to set up an appointment for a student.
    return HttpRequest("Not Implemented")

def register(request):
    # student's registration page
    # renders the student registration page for GET request, creates a new user for POST
    request if data is valid
    return HttpRequest("Not Implemented")

def student_profile(request):
    # present a student profile as specified in the wireframe, and a button to edit the profile
    on top right of the screen
    return HttpRequest("Not Implemented")

```

```

def edit_student_profile(request):
    # editing the student profile action, all of the information specified in the Student model
    with blank=True (a Student can stay anonymous as much as he likes)
    return HttpResponseRedirect("Not Implemented")

def counselor_list(request):
    # display a list of counselors with information about them, so the student can choose if
    he wants to change a counselor
    return HttpResponseRedirect("Not Implemented")

def student_tasks(request):
    # using the game API, a student will get a list of the task (exercises) that were assigned
    to him, and he either completed or not. He can continue any uncompleted assignment, while will
    lead him to the run_task() action
    return HttpResponseRedirect("Not Implemented")

def run_task(request, task_id):
    # using the game API, fetch the task from the game app and load it in the view for the
    student to interact with it
    return HttpResponseRedirect("Not Implemented")

## Counselor actions

def counselor_profile(request, counselor_id):
    # information specified in the wireframe and model that the counselor can
    edit_counselor_profile
    # in addition, this action renders information about the students he is with session with,
    what exercises did he give them, and a graphic schedule where the counselor can see all his
    meetings
    return HttpResponseRedirect("Not Implemented")

def counselor_profile(request, counselor_id):
    return HttpResponseRedirect("Not Implemented")

def manage_student_tasks(request, student_id):
    # this actions uses the Game API to get all the tasks assigned to a student and their
    status, and in addition using a POST request we can assign or change current assignments for
    a students and again using the game API modify these settings in the game app.
    return HttpResponseRedirect("Not Implemented")

## Stretch goals - Student ##

```

```
#def past_meetings(request): # need to add wireframe
#     # see all the past meetings the student had, pressing a meeting will lead to
meeting_details() action to see information about that meeting
#     return HttpResponseRedirect("Not Implemented")

#def meeting_details(request, meeting_id): # need to add wireframe
#     # information about a meeting the student had in the future - who was it with, when, what
assignments were assigned to the student in that meeting, and a text box where the counselor
can summarize the meeting and give tips and advices to the student.
#     return HttpResponseRedirect("Not Implemented")
```

References

1. Wireframing done via <https://app.moqups.com>
2. Graph image from statistics wireframe from
<http://www.asoprofessional.com/chapter-5-1-app-nametitle/>
3. Carnegie Mellon Logo from
<http://www.cmu.edu/marcom/brand-standards/logos-colors-type.html>