

# SKEDGE

**Smarter course scheduling for our  
University of Rochester**

Dan Hassin

Supervised by  
Professor Philip Guo

Department of Computer Science  
University of Rochester  
Rochester, New York

April 12, 2016

submitted in partial fulfillment of  
the requirements for the degree  
*honors bachelor of science*

# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Space of course explorers and schedulers . . . . .	1
1.2 Overview of CDCS . . . . .	1
1.3 Overview of Skedge . . . . .	1
<b>2 Design in reaction to CDCS</b>	<b>4</b>
2.1 Modernity . . . . .	4
2.2 Usability . . . . .	6
2.3 Search . . . . .	8
2.4 Social . . . . .	9
<b>3 Technical overview</b>	<b>11</b>
3.1 Back-end . . . . .	11
3.2 Front-end . . . . .	11
3.3 Analytics . . . . .	11
<b>4 Data analytics</b>	<b>12</b>
4.1 Usage . . . . .	12
4.2 Navigations-per-add . . . . .	13
4.3 Users' search types over time . . . . .	14
<b>5 Looking forward</b>	<b>15</b>
5.1 Features . . . . .	15
5.2 Analytics . . . . .	15
<b>6 Conclusions</b>	<b>16</b>
6.1 Proposal to the University . . . . .	16
6.2 Resources . . . . .	16
<b>Bibliography</b>	<b>17</b>

# List of Tables

# List of Figures

1.1	Better CDCS, a separate browser extension that embeds buttons into the CDCS course results interface, allowing users to add courses to a locally-stored schedule	2
1.2	CDCS and Skedge for the search query <code>csc</code>	3
2.1	CDCS in July 2, 2010, courtesy of <i>Archive.org</i>	5
2.2	CDCS and Skedge running on a mobile device	6
2.3	Section presentation on CDCS and Skedge	7
2.4	Hoverable and clickable course and instructor mentions in Skedge	7

## **Abstract**

In this paper I present Skedge, a web application for students to comfortably and effectively engage with the University’s course catalog. Skedge matches and surpasses the capabilities of the existing University tool for this purpose, “Course Description / Course Schedule” (CDCS) and presents its information in a more visually pleasing way. As a result, Skedge boasts strong user-retention rates, long session durations, and high student adoption despite having virtually no advertisement. Through collected usage data, I demonstrate that a) Skedge’s differences from and additions to CDCS are usable and have real need, b) the two major use-cases associated with course browsing—direct search and exploratory search—are effectively accommodated by Skedge, and c) Skedge’s search mechanism is user-friendly and self-teaches to users over time.

# Chapter 1

## Introduction

This paper will begin by

### 1.1 Space of course explorers and schedulers

### 1.2 Overview of CDCS


#### 1.2.1 “Better CDCS”

### 1.3 Overview of Skedge

Skedge is a website I developed in 2014 and have been maintaining and developing since.

1.2b





COURSE DESCRIPTION  
COURSE SCHEDULE

Course Description / Course Schedule (CDCS)

**Search**

Year/Term: (Required)  
Fall 2016

School:

Subject:

Course: (MTH 162 or MTH or 162)  
csc

Course Type:

Status:

Description - Keywords:

Title:

Instructor Name:

Day: ☐ M ☐ T ☐ W ☐ R ☐ F ☐ S ☐ U

Start Time Between:  and

Credit Hours Between:  and

Division:

☐ Upper Level Writing Courses

☐ Diversity Courses

Arts, Sciences, and Engineering      Computer Science

CRN	Course	Course Title	Term	Credits	Status
49714	CSC 131	RECREATIONAL GRAPHICS	Fall 2016	4.0	Closed

**Schedule:** Day: TR      Begin: 1105      End: 1220      Building: HARK      Room: 114

**Enrollment:** Section Enroll: 28      Section Cap: 28

**Instructors:** PAWLICKI T

**Prerequisites:** None

**Description:** A hands on introduction to 3D computer graphics and animation techniques taught from a user point of view. Topics include 3D modeling, animation, and simulation. Assessment based on projects. No written exams.

**Clusters:** N4CSC013, N4CSC014

CRN	Course	Course Title	Term	Credits	Status
49723	CSC 161	INTRO TO PROGRAMMING	Fall 2016	4.0	Open

**Schedule:** Day: TR      Begin: 1815      End: 1930      Building: B&L      Room: 109

**Enrollment:** Section Enroll: 32      Section Cap: 115

**Class Info:** YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION.

**Instructors:** SARKIS R; STJACQUES R

**Prerequisites:** None

**Description:** Organized thinking, creative problem solving, and the precise description of solutions are valuable skills in academia and life. The formulation and solution of problems using computers is increasingly important in all artistic and scholarly fields. We introduce core concepts and techniques of programming as a way to develop these skills, as basis for further CS study, and for application to other fields. Lab required.

**Clusters:** N4CSC013, N4CSC015, N4CSC016, N4CSC017, N4CSC018, N4CE001

CRN	Course	Course Title	Term	Credits	Status
78686	CSC 161	INTRO TO PROGRAMMING	Fall 2016	4.0	Open

**Schedule:** Day: MW      Begin: 1650      End: 1805      Building: GRGEN      Room: 101


**Enrollment:** Section Enroll: 53      Section Cap: 115

**Class Info:** YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION.

**Instructors:** STJACQUES R; SARKIS R

**Prerequisites:** None

(a) CDCS



Choose a schedule:

Spring 2016

Fall 2016

**CSC 131: Recreational Graphics** 4 credits

A hands on introduction to 3D computer graphics and animation techniques taught from a user point of view. Topics include 3D modeling, animation, and simulation. Assessment based on projects. No written exams.

Add Section

**Time & Place:** Tues/Thurs 11:05am-12:20pm, HARK 114  
**Instructor:** Pawlicki T      **CRN:** 49714

28/28 enrolled

**CSC 161: Intro to Programming** 4 credits

**Comments:** YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION.

Organized thinking, creative problem solving, and the precise description of solutions are valuable skills in academia and life. The formulation and solution of problems using computers is increasingly important in all artistic and scholarly fields. We introduce core concepts and techniques of programming as a way to develop these skills, as basis for further CS study, and for application to other fields. Lab required.

Add Section

**Time & Place:** Tues/Thurs 6:15pm-7:30pm, B&L 109  
**Instructor:** Sarkis R, St Jacques R      **CRN:** 49723

32/115 enrolled

Add Section

**Time & Place:** Mon/Wed 4:50pm-6:05pm, GRGEN 101  
**Instructor:** St Jacques R, Sarkis R      **CRN:** 78686

53/115 enrolled

[Show 6 lab sections](#)

	M	T	W	R	F
12					
1					
2					
3					
4					
5					
6					
7					
8					

REL 389W

MTH 165

MTH 165

2 sections / 8 credits

(b) Skedge

Figure 1.2: CDCS and Skedge for the search query csc



## Chapter 2

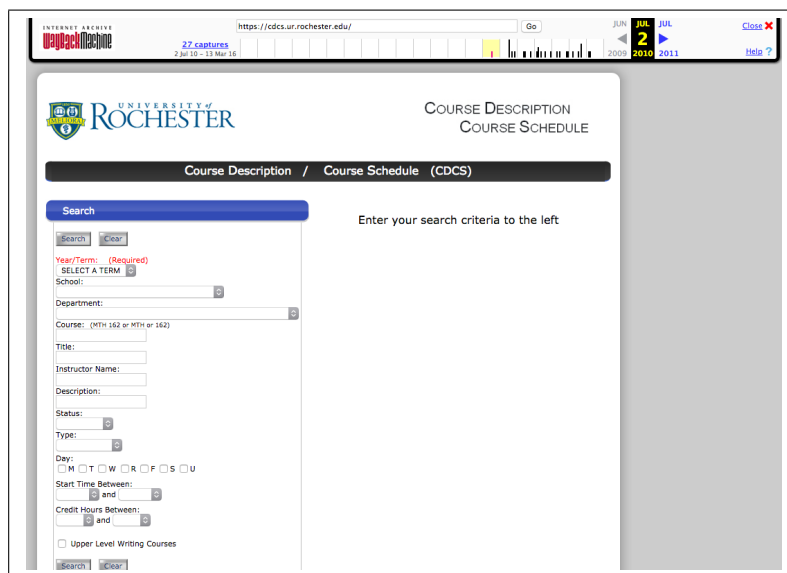
# Design in reaction to CDCS

From its very inception, Skedge’s functionality and visual design were driven by the shortcomings of CDCS. It was built *bottom-up*, not *top-down*—every aspect of the application was either made as a reaction to a particular grievance in CDCS or as the natural evolution of an existing feature. Skedge is thus rooted in *usability* derived from real need, not mere conjecture along the question “what could students want?”. Its success with students, shown in Chapter 4, demonstrates that this usability extends beyond my own standard and can fulfill the various discovered use-cases of students in general.

In this chapter, I will

### 2.1 Modernity

CDCS is an old system, relatively speaking, and its recent development has been stagnant. It launched in 2009, seven years ago, and has hardly changed since. Figure 2.1 shows CDCS in July 2010, which, besides the addition of a few search fields, is identical to its current version. Since its introduction in 2009, we have seen the rise of mobile devices into ubiquity, a boom in “hacker culture” and public APIs, and the capability for standalone web applications to be as sophisticated and dynamic as desktop-class applications without the aid of browser extensions.



**Figure 2.1:** CDCS in July 2, 2010, courtesy of *Archive.org*

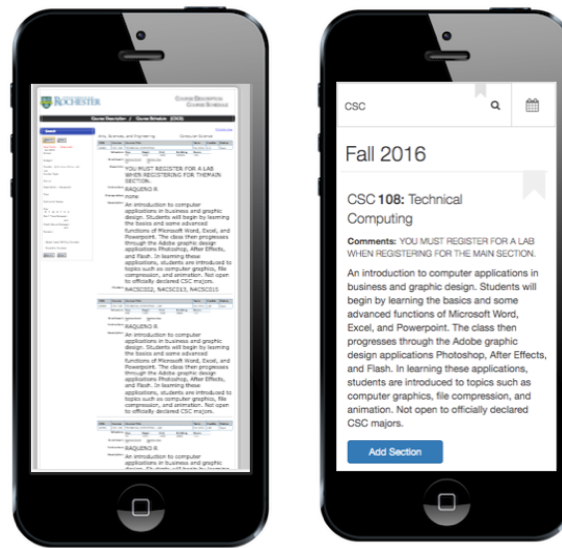
### 2.1.1 Mobile

- Important nowadays
  - Mobile traffic stat or smth

### 2.1.2 Public API

With the increasing number of attendees at University of Rochester’s hackathons, it is clear that the University’s “hacker culture” is growing (e.g. more students collaborating to build side-projects that integrate resources and services often benefitting the student community), and projects that include the University’s course catalog are becoming more common. Open-source and open-information services greatly help to foster such innovation, and having public APIs is essential toward this end.

Skedge provides a public JSON API at the root URL <http://www.skedgeur.com/api/>, made at the request of a student that was interested in using its course data, and the API has already been used in projects by several other student groups. The endpoints included are `/api/courses?q=query` (Skedge’s query language—described in detail in section 2.3—is supported here), `/api/departments?q=optional_query`, and `/api/instructors?q=optional_query`.



(a) CDCS

(b) Skedge

**Figure 2.2:** CDCS and Skedge running on a mobile device

### 2.1.3 Built-in scheduler vs. browser extension

- Better UX
  - Data is centralized

### 2.1.4 GET requests vs. AJAX

- Can use back button
  - Can send a link to a course or search

## 2.2 Usability

### 2.2.1 Data quality

- Courses don't shout - Typos in comments - 12-hour time

### 2.2.2 Section display

- Grouped course sections - Embedded labs (A/B too), workshops, & recitations

- More concise, space efficient (collapsible) - Eliminates redundancy of same stuff (instructor, description, course title, etc) - Don't have to pay more attention that you should to whether it's a different course or the same one

Note that in Figure 2.3a, the first two boxes are sections for the same course, and the next two are labs for that course. Four more lab sections and *twenty* more workshop sessions for that same course follow below the truncated screenshot. Figure 2.3b demonstrates how this information can be conveyed more concisely.

CRN	Course	Course Title	Term	Credits	Status
49723	CSC 161	INTRO TO PROGRAMMING	Fal 2016	4.0	Open
<b>Schedule:</b> Day: TR, Begin: 18:15, End: 19:30, Building: B&L, Room: 109 <b>Enrollment:</b> Section: 53, Seats: 113 <b>Class Info:</b> YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION. <b>Instructor:</b> SARKIS R, STACQUES R <b>Prerequisites:</b> None <b>Description:</b> Organized thinking, creative problem solving, and the precise description of solutions are valuable skills in academia and life. The formulation and solution of problems using computers is increasingly important in all artistic and scholarly fields. We introduce core concepts and techniques of programming as a way to develop these skills, as basis for further CS study, and for application to other fields. Lab required. <b>Clusters:</b> WICKC013, WICKC015, WICKC016, WICKC017, WICKC018, WICKC001					
78686	CSC 161	INTRO TO PROGRAMMING	Fal 2016	4.0	Open
<b>Schedule:</b> Day: MW, Begin: 16:50, End: 18:05, Building: GRGEN, Room: 101 <b>Enrollment:</b> Section: 53, Seats: 113 <b>Class Info:</b> YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION. <b>Instructor:</b> STACQUES R, SARKIS R <b>Prerequisites:</b> None <b>Description:</b> Organized thinking, creative problem solving, and the precise description of solutions are valuable skills in academia and life. The formulation and solution of problems using computers is increasingly important in all artistic and scholarly fields. We introduce core concepts and techniques of programming as a way to develop these skills, as basis for further CS study, and for application to other fields. Lab required. <b>Clusters:</b> WICKC013, WICKC015, WICKC016, WICKC017, WICKC018, WICKC001					
49737	CSC 161	INTRO TO PROGRAMMING LAB	Fal 2016	LAB	Open
<b>Schedule:</b> Day: MW, Begin: 18:15, End: 19:30, Building: GAVET, Room: 244 <b>Enrollment:</b> Section: 53, Seats: 50 <b>Class Info:</b> YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION. <b>Instructor:</b> SARKIS R, STACQUES R <b>Prerequisites:</b> None <b>Description:</b> Organized thinking, creative problem solving, and the precise description of solutions are valuable skills in academia and life. The formulation and solution of problems using computers is increasingly important in all artistic and scholarly fields. We introduce core concepts and techniques of programming as a way to develop these skills, as basis for further CS study, and for application to other fields. Lab required. <b>Clusters:</b> WICKC013, WICKC015, WICKC016, WICKC017, WICKC018, WICKC001					
49818	CSC 161	INTRO TO PROGRAMMING LAB	Fal 2016	LAB	Open

(a) Ungrouped sections in CDCS

**CSC 161: Intro to Programming** 4 credits

**Comments:** YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION.

**Remove Section** **Time & Place:** Tues/Thurs 6:15pm-7:30pm, B&L 109  
**Instructor:** SARKIS R, St Jacques R **CRN:** 49723

**Add Section** **Time & Place:** Mon/Wed 4:50pm-6:05pm, GRGEN 101  
**Instructor:** St Jacques R, SARKIS R **CRN:** 78686

**Lab sections:** (hide)

**Add** **Time & Place:** Mon/Wed 6:15pm-7:30pm, GAVET 244  
**Instructor:** SARKIS R, St Jacques R **CRN:** 49737

**Remove** **Time & Place:** Tues/Thurs 2:00pm-3:15pm, HARK 114  
**Instructor:** SARKIS R, St Jacques R **CRN:** 49820

**Add** **Time & Place:** Mon/Wed 12:30pm-1:45pm, GAVET 244  
**Instructor:** SARKIS R, St Jacques R **CRN:** 83575

**Add** **Time & Place:** Mon/Wed 12:30pm-1:45pm, GAVET 244  
**Instructor:** SARKIS R, St Jacques R **CRN:** 83581

**Show 20 workshop sections**

(b) Grouped sections in Skedge

Figure 2.3: Section presentation on CDCS and Skedge

## 2.2.3 Course reference

**CSC 172: Data Structures and Algorithms**

**Comments:** YOU MUST REGISTER FOR A LAB & WORKSHOP WHEN REGISTERING FOR THE MAIN SECTION.

**Prerequisites:** [CSC 171](#)

**Add Section** **Time & Place:** Tues/Thurs 2:00pm-3:15pm, HARK 114  
**Instructor:** SARKIS R, St Jacques R **CRN:** 49820

**Show 6 lab sections**

**Show 19 workshop sections**

**CSC 171: Introduction to Computer Science** 4 credits  
**Description:** Discovering, formulating, and exploiting the structure of problems to aid in their solution by computer. An introduction to algorithmic problem solving and computer programming...

Figure 2.4: Hoverable and clickable course and instructor mentions in Skedge

- Clickable/hoverable course links, professor searches

### 2.2.4 Multiple schedule support

- Old CDCS+betterCDCS system can't keep track of this, have conflicts when adding stuff

### 2.2.5 Exporting to GCal, .ics, image

- Mobile sync support - Security: BetterCDCS export gcal is currently broken and sends netID in PLAINTEXT over http(!!!)

### 2.2.6 Search

Most important usability concern is finding courses.

## 2.3 Search

Use cases, natural language.

### 2.3.1 Course selection criteria

Narrowed it down to three criteria. Keep in mind that *none* of the things listed below are supported by CDCS, and they are all supported by Skedge.

#### Requirements

- Finding crosslists - Clusters

#### Browsing

- “New” courses - “Autofit” search - Random - Sorts

#### Friends

- “What are my friends taking?” (“what are you taking this semester” = probably most common smalltalk phrase uttered on campus) - “What do my friends recommend?” - “have you taken this class, and if so, what did you think of it?”

### **2.3.2 Natural language search**

See figure.

#### **Advantages**

- 15 fields reduced to 1
- vs form entry: - Faster - More intuitive - More easily extendable

#### **Disadvantages**

Having to know the DSL, grammar ambiguities (can be solved with a ‘did you mean’)

### **2.3.3 Multipurpose**

Used by other links (instructors, course references) around the site

### **2.3.4 Added features**

- CRN (!) - Crosslist - Class size

## **2.4 Social**

### **2.4.1 The issue**

#### **Static image vs. live site**

- Edits don’t update - Referencing courses

#### **Finding common courses**

- requires your friends to share their schedules on FB publicly and you to see their post
  - is schedule-first, not search-first - typically only occurs for the current semester

### **2.4.2 Skedge Social**

**Friends' course enrollments**

Mini-feed

**Friends' course likes**

**Likes & enrollments embedded in results**

**Personal schedule synchronization**

**Privacy**

**Notifications**

## Chapter 3

# Technical overview

### 3.1 Back-end

Skedge's infrastructure is built

*nginx, unicorn, Ruby on Rails, PostgreSQL, React.js, Ahoy, and Google Analytics.*

### 3.2 Front-end

### 3.3 Analytics



# Chapter 4

## Data analytics

Hypotheses:

1. Skedge's differences from and additions to CDCS are usable and have real need
2. Skedge's navigations-per-add and other metrics demonstrate effectiveness of the use cases  
a) direct searching, and b) course browsing
3. Skedge's DSL is user-friendly; users learn more advanced search types over time by using it

### 4.1 Usage

#### 4.1.1 General

Since November 3rd 2015 (137 days) 3,768 unique users 4,500 schedules Average 90 sessions/day  
Average 4.92 pages/session Average 5:31 minutes/session 28% of sessions are from new users

MOBILE RESULT

#### 4.1.2 Search

**Empty searches**

Can learn from these Some funny ones

### 4.1.3 Course blocks

40% of sessions have at least one block-click Average of 4.94 block-clicks per session

### 4.1.4 Social

90 users have linked Skedge to Facebook Since March 1st, 4,000+ visits (200 visits/day) 60% of visits to /social were returning visitors 90 overlays onto friends' schedules 10 clicks to Facebook profiles :( - get stats from the fb dashboard

### 4.1.5 Conclusion

Success! Considering skedge is OPTIONAL. + course blocks (obv usecase, can't click) + exports (not supported by thing) + mobile

## 4.2 Navigations-per-add

### 4.2.1 Definitions

A navigation is defined as a search, or a click on an instructor's name, or a click on a crosslisted or prerequisite course link

The navigations-per-add, bookmark measure is the number of navigations a user took (within one session) until a course was added, bookmarked

### 4.2.2 Trends

### 4.2.3 Breaking them apart

behavioral patterns Direct search for specific course Discovery, browsing, exploring

**Direct searches**

**Browse**

### 4.2.4 Conclusion

Effective++

## 4.3 Users' search types over time

### 4.3.1 Definitions

Points for search by (omits number and dept.):

description credits crosslisted CRN instructor title year term 'random' upper-level writing  
"CSC" 0 "MTH 165" 0 "taught by hema" 1 (2 searches) "random mur 1-2 credits" 2 (1  
search)

### 4.3.2 Trends

First increase (60.5Median: 2 searches Average: 4.23 searches (Starting at 1 counts as an increase  
value of 0)

Second increase (7.9Median: 8 searches Average: 17.52 searches

### 4.3.3 Conclusion

DSL++

## Chapter 5

# Looking forward

### 5.1 Features

### 5.2 Analytics

## Chapter 6

# Conclusions

### 6.1 Proposal to the University

### 6.2 Resources

#### Source code

The source code for Skedge is available online under an open source license:

<https://github.com/RocHack/skedge>.

#### Live site

The site can be found at: <http://skedgeur.com>.

# Bibliography

- [1] Takis Konstantopoulos *Introductory lecture notes on Markov Chains and Random Walks*.  
Uppsala University,  
<http://www2.math.uu.se/~takis/L/McRw/mcrw.pdf>

## Appendix