Syllabus

Catalog Description

IT 390: Rapid Development of Scalable Cloud Applications (3 credit).

Presents software engineering, programming techniques, platforms and tools necessary for rapid development of scalable applications, including cloud platforms; scalable data storage solutions; web applications development environments. The course will provide a general overview of such techniques but will concentrate on selected ones in each term. The students will work in small teams and must develop scalable prototypes during the course.

Prerequisites

Prerequisites: IT 206^a (or IT 209^b, CS 211^b) and IT 213^c (or IT 193^c) and IT 214^c (or IT 194^c)

Expected Outcomes

The outcomes expected for a student passing this course are:

- Understand the pros and cons of rapid application development (RAD)
- Understand various strategies for rapid application development (e.g. Agile, Extreme, Joint, Lean, Scrum, Spiral)
- Understand the role of scalability in RAD and available solutions (e.g. cloud solutions from Google, Amazon, Microsoft)
- Understand the advantages and disadvantages of using platform independent data storage techniques (e.g. Java Data Objects, Java Persistence API, MySQL) and platform dependent data storage techniques (e.g. Google Datastore)
- Proficiently use of client-side programming (e.g. CSS, HTML, JavaScript, AngularJS, Angular).
- Proficiently use the Java or Python programming language within the strategies for RAD.
- Create a rapid prototype for the user interface of an IT application
- Work within a team to develop a prototype for an IT application

Supported Student Outcomes at the Program Level

(1) An ability to analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
(2) An ability to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
(3) An ability to communicate effectively in a variety of professional contexts.
(5) An ability to function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.
(6) An ability to identify and analyze user needs and to take them into account in the selection, creation, integration, evaluation, and administration of computing-based systems.
Section

The course has one section IT 390-001 (CRN: 21362)

- Class lecture
- Wednesday 4:30 - 7:10
- PW-KJH 248

Course EMAIL Communication

When you use email communication in the course it is important to follow the rules below:

- **IN BLACKBOARD:**
  - Do not use ALL INSTRUCTORS or ALL GTAS option as there are other persons recorded on such positions

- **OUTSIDE BLACKBOARD:**
  - SUBJECT: Start with "IT390-001" then follow with the issue
  - CC: all responsible parties (i.e. for a GTA grading issue that you escalate to instructor, include the GTA)

Course Instructor and Office Hours

Course Instructor: Dr. Can X. Nguyen.
See Contact Information on Blackboard for Office Hours.

GTA

Course GTA: Pranavi Nallamalla.
See Contact Information on Blackboard for Office Hours.

Course Coordinator

Dr. Mihai Boicu

- You must contact the course coordinator only after you contacted and tried to resolve an issue with your course instructor and/or assigned GTA.
- You may contact the course coordinator for general feedback related to the course or for specific complains.
- Phone: (703) 993-1591 (M-F 9AM-3PM)
- Email: mboicu@gmu.edu (start email subject with IT390-section COORDINATOR)
- Office hours by appointment, send me 5 large time intervals.
## Fall 2021 Schedule

The weekly session will combine practical skills, analytical skills and theoretical knowledge through creative team working. We will cover theoretical concepts, available tools and methods and project guidance and development. Also, you will have readings, independent and team system development.

The following is a sample list of topics but their organization per weeks may vary. However, based on the specific project(s) selected the topics and their order may vary. New topics may be added. Some topics may be just briefly presented.

A detailed description for each week will be published in the weekly folder. At the beginning of the course the focus will be more on the theoretical aspects and toward the end will shift to the project finalization.

### Rapid Application Development

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to Rapid Application Development (System Lifecycle, Agile Development)</td>
</tr>
<tr>
<td>2</td>
<td>Initiating a Project (Agile user stories, Agile Lifecycle, Justifying a Project, Stakeholders, Architecture)</td>
</tr>
<tr>
<td>3</td>
<td>Product Owner (Role, Tasks and Requirements)</td>
</tr>
<tr>
<td>4</td>
<td>User stories</td>
</tr>
<tr>
<td>5</td>
<td>Architecture Owner</td>
</tr>
<tr>
<td>6</td>
<td>Iteration Modeling, Design Sprints</td>
</tr>
</tbody>
</table>

### Google Cloud Elements

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computational platform: Google App Engine (GAE)</td>
</tr>
<tr>
<td>2</td>
<td>Databases: Google Datastore (Firestore)</td>
</tr>
<tr>
<td>3</td>
<td>Cloud Services (maps, identity management)</td>
</tr>
</tbody>
</table>

### Cloud and Web Programming

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Java Servlets, JSP</td>
</tr>
<tr>
<td>2</td>
<td>Type script and Angular</td>
</tr>
</tbody>
</table>
## PROJECT

1. Topic discussion; Team forming; Team contract, Project Charter
2. Initiate an Agile Project
3. Product Owner, User stories, Architecture Owner, Initial Modeling, Back-end management
4. Alpha version of user story 1
5. Beta version of user story 1
6. Final version and presentation of user story 1 implementation
7. Revise user stories and select user story 2, initial modeling, sessions
8. Alpha version of user story 2
9. Beta version of user story 2
10. Final presentation of user story 2
11. Revise user stories, select user story 3, initial modeling
12. Alpha version of user story 3
13. Beta version of user story 3
14. Final deployment of the application
15. Final Report; Public Presentation;
# IT 390: Rapid Development of Scalable Cloud Applications

**Syllabus**  
**Fall 2021**

<table>
<thead>
<tr>
<th>THEORY</th>
<th>TOOLS AND METHODS</th>
<th>PROJECT</th>
</tr>
</thead>
</table>
| Rapid Application Development Methods | Project Management Tools (code and documentation)  
Google App Engine (GAE) overview | Topic discussion; Team forming; Setup of development platform |
| Requirements Planning; User centric design | Interaction design tools  
Interface design tools; GAE applications | Initial project planning for 2 phases; Detailed Design for phase 1 (interaction and interface) |
| Rapid development of web applications – client side | Google Web Toolkit (GWT) Overview  
GWT Designer Overview | Client demo design and implementation |
| Rapid development of web applications – server side | Java Servlets, Web API and Microservices | Server demo design and implementation |
| Application life-cycle | GAE applications | Interaction demonstration |
| Session life cycle | GAE Datastore; Objectify; | Design; Data Model |
| Distributed Data Storage | Building a data model | Server implementation |
| High Replication Storage | Java Data Objects (JDO) | Services implementation |
| Storage Cost | GAE native data store; SQL Option | Client implementation |
| Application portability | Amazon, Azure, Google clouds | Minimal interaction prototype |
| Application security | Encryption; Input validation; | Improve functionality |
| User privacy | Test units | Enhancing user interaction |
| Application testing | Clouds Application Testing | Final testing |
| Application deployment; Versioning | Clouds DevOps | Final deployment |

Additional special topics might be addressed during the course, including: distributed computing with large data sets (e.g. Hadoop MapReduce, SPARK), data interchange formats (e.g. JSON), data modeling formats (e.g. OWL, RDF).
Online Textbooks and Documentation:

This is a list of online documentation and books. However, specific reading will be provided each week, based on the weekly portfolio.

4. Instructor course notes (they include a detailed tutorial for the project example with explanations)
5. Google App Engine Tutorials and Documentation (https://cloud.google.com/appengine/docs/)
8. AngularJS Tutorial: https://docs.angularjs.org/tutorial
Grading

The students will be assigned a grade as follow:

- Weekly Project assignments (42%)
- Class participation and project team contribution (8%)
- Project first use case – week 7 (10% - including class presentation; draft report; code and demo)
  - The presentation is a description of the project in class
  - The demo consists of a successfully deployed application and the source code
- Project second use case – week 11 (10% - including class presentation; draft report; code and demo)
  - The presentation is a description of the project in class
  - The demo consists of a successfully deployed application and the source code
- Final project delivery – week 15 (30% including class presentation; draft report; code and demo)
  - The final report must be a team paper
  - At the end of the report each student must specify her or his detailed contributions to the project.

The grading scale for this course is:

<table>
<thead>
<tr>
<th>Numeric Grade</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>97 – 100%</td>
<td>A+</td>
</tr>
<tr>
<td>93 – 96%</td>
<td>A</td>
</tr>
<tr>
<td>90 – 92%</td>
<td>A-</td>
</tr>
<tr>
<td>87 – 89%</td>
<td>B+</td>
</tr>
<tr>
<td>83 – 86%</td>
<td>B</td>
</tr>
<tr>
<td>80 – 82%</td>
<td>B-</td>
</tr>
<tr>
<td>77 – 79%</td>
<td>C+</td>
</tr>
<tr>
<td>73 – 76%</td>
<td>C</td>
</tr>
<tr>
<td>70 – 72%</td>
<td>C-</td>
</tr>
<tr>
<td>60 – 69%</td>
<td>D</td>
</tr>
<tr>
<td>0 – 59%</td>
<td>F</td>
</tr>
</tbody>
</table>
Hardware and Software requirements

For all sections you must have a personal computer with internet connection and webcam. It is strongly recommended that you have a powerful enough laptop on which you can perform code development that you can bring to class.

For the online section you must have a personal computer with internet connection, with speakers and microphone.

We require either a Mac OS X or Windows 10 computer.

Course Delivery Methods

The course will be delivered using various methods. You must have your MASON email account activated and you must check your email daily for announcements related to the course. You must have access to Blackboard Learning System and to know how to use its features.

There are video presentations posted on the Blackboard. You must have an environment in which you can watch these videos.

You will have several assignments and assessments to be performed each week. A summary of weekly requirements will be sent at the beginning of the week.

COURSE CANCELED (SNOW DAYS)

If the courses are canceled the first option is to have a synchronous meeting online during the same times. If you cannot be online the course will be recorded and posted on the course Blackboard site.

Intellectual Property

There is a strong recommendation that all work in the class projects to be done based on an open source license (e.g. Academic Free License http://en.wikipedia.org/wiki/Academic_Free_License). This will allow a rich, shared exchange of ideas and will allow each member of the class to further benefit with no restriction from the work performed in the class.

Privacy

Instructors respect and protect the privacy of information related to individual students. Specific issues relating to an individual student will be discussed via email, telephone or in person. Instructors will not discuss issues relating to an individual student with other students (or anyone without a need to know) without prior permission of the student. There is no guarantee related to the security of email and telephone conversations.

Assessable work other than final exams will be returned to individual students directly by the Instructor (or by a faculty or staff member or a Teaching Assistant designated by the Instructor, or via another secure method). Under no circumstances will a student’s graded work be returned to another student.
Faculty and staff will take care to protect the privacy of each student’s scores and grades.

Because of the nature of this class, some work performed by the student will be published and discussed in the class. Other students will be able to make comments and suggestions related to the published work, without seeing the actual grade the student earned for the work.

Disability Accommodations

The Office of Disability Services (ODS) works with disabled students to arrange for appropriate accommodations to ensure equal access to university services. Any student with a disability of any kind is strongly encouraged to register with ODS as soon as possible and take advantage of the services offered.

Accommodations for disabled students must be made in advance – ODS cannot assist students retroactively, and at least one week's notice is required for special accommodations related to exams. Any student who needs accommodation should contact the Instructor during the first week of the semester so the sufficient time is allowed to make arrangements.

Honor Code

All members of the Mason community are expected to uphold the principles of scholarly ethics. Similarly, graduating students are bound by the ethical requirements of the professional communities they join. The ethics requirements for some of the communities relevant to Applied IT graduates are available via the following links:

- ACM Code of Ethics and Professional Conduct
- IEEE Code of Ethics
- EC-Council Code of Ethics

On admission to Mason, students agree to comply with the requirements of the Mason Honor System and Code\(^1\). The Honor Code will be strictly enforced in this course. Honor Code cases are heard by a panel consisting of students – students who meet the requirements are encouraged to nominate themselves to serve on the Honor Committee.

Any use of the words or ideas of another person(s), without explicit attribution that clearly identifies the material used and its source in an appropriate manner, is plagiarism and will not be tolerated. There is a "zero tolerance" policy for plagiarism within The Volgenau School. The Instructor reserves the right to use manual and/or automated means (including such services as Turnitin.com) to detect plagiarism in any work submitted by students for this course, and to direct Teaching Assistants and/or other faculty and/or staff members to do likewise in support of this course.

For this course, the following requirements are specified:

\(^1\) Available at [http://catalog.gmu.edu](http://catalog.gmu.edu) and related Mason Web pages.
All assessable work is to be prepared by the individual student, unless the Instructor explicitly directs otherwise.

All work must be newly created by the individual student for this course for this semester. Any usage of work developed for another course, or for this course in a prior semester, is strictly prohibited without prior approval from the instructor.

For team work a summary at the end of the submission must identify mutually agreed individual contributions.

Students may seek assistance with assigned work (and are encouraged to do so if they feel the need), provided:

• The directions for the assigned work do not prohibit such assistance.

• Such assistance is acknowledged in the submitted work, clearly identifying the person(s) giving assistance and the nature of the assistance given.

• Any work to be submitted is prepared entirely an exclusively by the student submitting it. Students are expressly prohibited from sharing any assessable work for this course in any manner with other students (except students assigned as Teaching Assistants or Undergraduate Peer Mentors to this course and the student’s section), unless all students involved have had their work graded and returned by the Instructor, or the Instructor has explicitly approved such sharing.