Следим за прогрессом здесь<https://docs.google.com/spreadsheets/d/1BjWZCSejIwqT4YVPm6seWRpZafW2mhBwZhsgFTznK1k/edit#gid=0>

Для зачета Вами (**ИНДИВИДУАЛЬНО** - групповые работы не засчитываются, будет проверка на антиплагиат и количество коммитов должно быть больше 15) должен быть реализован проект по теме. Тему необходимо описать (раскрыв что вы собираетесь делать) и согласовать, написав:

1. Что вы хотите сделать (кратко - до 3х предложений)
2. На каком языке(ах) будет выполнен проект и с какими технологиями (до 3х предложений)
3. Каким функционалом будет обладать решение в итоге, что он будет уметь и зачем это нужно этому проекту уметь (до 180 слов)

Тема проекта может быть связана с тематикой диплома или другими курсами, но проект должен быть полностью реализован до даты сдачи зачета. У проекта на GitHub должно быть ридми содержащее помимо прочего фразу:

*Project for Grid & Cloud course on the Faculty of Applied Mathematics and Control Processes of Saint Petersburg University 2020*

Для зачета любая тема должна быть раскрыта в виде решения трех задач с размещением исходного кода на GitHub (в случае когда задачи решены не для работодателя).

1. База даннных (ORM) + REST API к ней + Бизнеc логика (любой язык)
2. Веб морда\мобильное приложение на Xamarin приложение к ней (использующая готовый интерфейс из<https://github.com/DesignRevision/shards-dashboard> или <https://github.com/Baseflow/XF-Material-Library> или <https://github.com/creativetimofficial/material-dashboard> ) общающееся по REST с сервисом
3. Система автоматизированной сборки и сбора информации о покрытии кода тестами (Code Coveredge от 50%) CI (AppVeyor (Windows) или TravisCI/CircleCI (Linux/Mac))

Для получения дополнительных баллов при сдаче экзамена вам необходимо интегрировать\внедрить в рамках выбранной темы в проект один или несколько из следующих компонентов (фич, task) и обосновать его полезность:

1. P2P Брокер передачи сообщений<https://github.com/dltcspbu/YAMB>
2. Смоделировав работу вашей системы и создав набор тестов используя<https://github.com/dltcspbu/madt>
3. Создав скрипт развертки и тестирования проекта в Ansible
4. Интегрировав в решение вашей задачи сервис отображающий информацию на карте <https://github.com/dltcspbu/geo>
5. Брокер передачи сообщений RabbitMQ
6. ElasticSearch (можно использовать его как БД) (применив по крайней мере 2 его возможности из этих: полнотекстовый поиск по сущностям, подсказки ввода, аналитика)
7. Использовав для больших объемов мелких данных БД OmniSciDB способную применять при работе с запросами GPU
8. Распределенный Реестр Hyperledger Fabric (можно использовать его как БД) (применив по крайней мере 2 его возможности: консенсусное принятие решений, подписи документов)
9. Распределенный Реестр Corda (можно использовать его как БД) (применив по крайней мере 2 его возможности: консенсусное принятие решений, подписи документов)
10. Распределенный Реестр Neo (можно использовать его как БД) (применив по крайней мере 2 его возможности: консенсусное принятие решений, подписи документов)
11. Распределенный Реестр Exonum (можно использовать его как БД) (применив по крайней мере 2 его возможности: консенсусное принятие решений, подписи документов)
12. Распределенный Реестр bigchaindb (можно использовать его как БД) (применив по крайней мере 2 его возможности: консенсусное принятие решений, подписи документов)
13. Система синхронизации состояния Apache ZooKeeper
14. Система анализа потоков данных Apache Flink
15. Использовав систему *service mesh или Directory Service*
16. Создав для проекта написанного на С\С++ обертку для других языков на Simplified Wrapper and Interface Generator (swig)
17. ZKP алгоритм на базе <https://github.com/sikoba/isekai>
18. Систему передачи сообщений<https://github.com/w3f/messaging/>
19. Интегрировав визуализацию данных стороне [VTK](http://www.vtk.org/)/ITK (можно из веб версию, можно использовав рендеринг на серверной стороне)
20. Интегрировав визуализацию графов используя<https://github.com/Netflix/vizceral> или GraphViz (например этот<https://github.com/magjac/d3-graphviz> )