# **Help Fight COVID19** [a social web app]

Contributors: Helal Uddin (mhud@kth.se/sujon2100@hotmail.com)
Nick Steele (nsteele@kth.se/njws1)

DD2482: *Automated software testing and DevOps*, May 26, 2020

# Contents

# 1 Background

This project has been developed for gathering information regarding COVID. It is a not-for-profit project with an emphasis on novelty and helping people within the community by providing them with reliable information about COVID-19 through a web application. Both students and individuals from the community will be supported by the functionality that the web application provides. This web app is built using the Microsoft Cloud Platform, Azure DevOps and Azure Webs Services. As a result, we have achieved a low cost, highly available, and high performing web application.

## 1.1 Technologies Used

Our precise technical contributions have been made using the following technologies.
**Front-End:** HTML, CSS, Bootstrap.
**Back-End:** C-Sharp with Microsoft .Net Core framework.
**Development Pattern:** Model View Controller (MVC)
**CI/CD:** Azure DevOps.

## 1.2 Built-in DevOps Technology

For continuous delivery, deployment, and code management, Azure DevOps has been used together with Azure Web Services. This ensures the seamless integration from the beginning/planning phase to the delivery phase.

## 1.3 Azure DevOps for COVID-19 project

Azure DevOps project is created to use Team Foundation Services (TFS) and agile methodology to integrate and manage the development process from the Visual Studio Covid19 project. This could help to support the Covid19 affected community with reliable online corona-virus update.

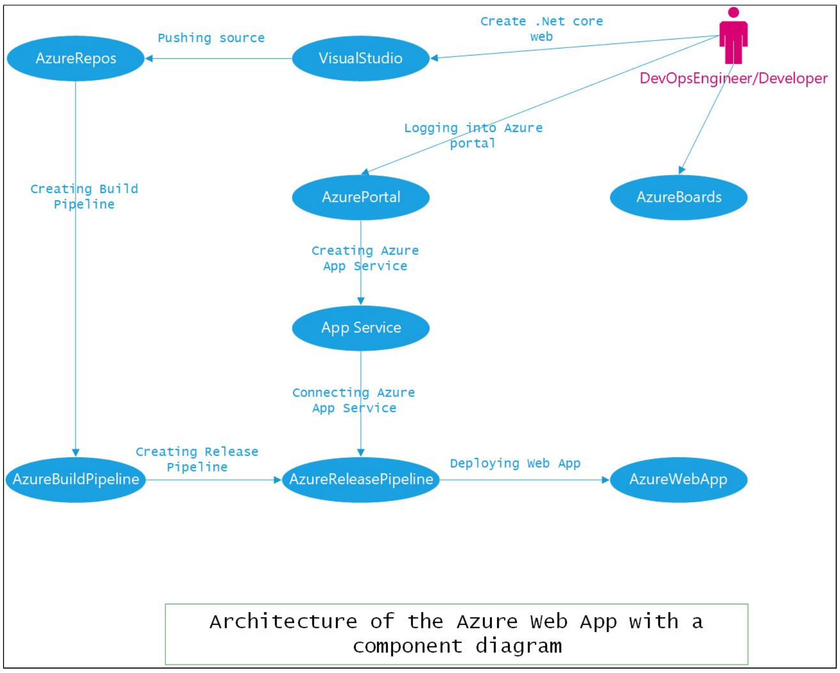
## 1.4 Azure Web Services

A Microsoft Azure service plan has been created with resource groups to run the local project in the cloud. Azure Web Service are used and configured within the Azure portal.

# 2 The Web Application

The COVID-19 ASP.NET Core web application is created using C, HTML, CSS code and Bootstrap to develop the Web Application. The application is built upon a Model-View-Controller, used to render data from the controller to the view. This helps to achieve separation of concerns and helps streamline the development structure so that work can be easily distributed to other developers in the future. Link to the source code: https://github.com/KTH/devops-course/tree/master/contributions/covid19/nsteele-mhud

### 2.1 Architectural Component View of the web app

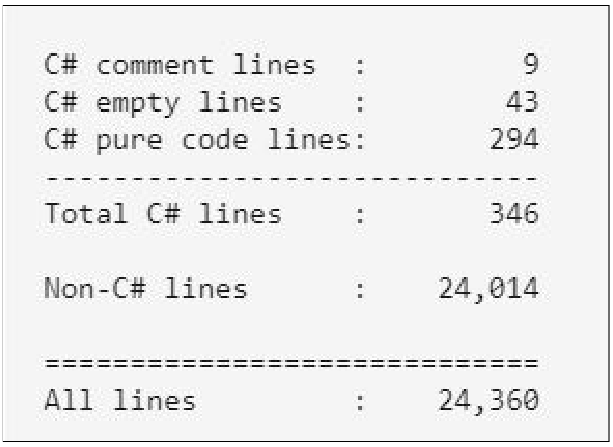


## 3 The test project

Inside of the Project Solution, a test project is added which would be expanded on in the future. A reference to the main project is added and all libraries have been installed from the NuGet Package Manager. Currently implemented is a test on the HomeController, its purpose is to check whether the index of the home controller is empty. Once the project is ready, a test run has been made and all the test parameters have been satisfied.

## 4 Lines of Code Metrics for the Web Application

Lines of Code (LOC) metric is used to count the total number of codes that are used in this project. To achieve this the Supercharger plugin is used. Supercharger allows us to count statistics for the entire project solution of the project, shown below is the statistics from our project.



```
C# comment lines  :        9
C# empty lines    :       43
C# pure code lines:      294
----------------------------
Total C# lines    :      346

Non-C# lines      :   24,014

============================
All lines         :   24,360
```

Figure: Overall Project LOC Metrics.

## 5 Code Coverage Metrics

| Hierarchy | Maintain -ability Index | Cyclomat -ic Complexity | Depth of Inheritance | Class Coupling | Lines of Source Code | Lines of Executable code |
|---|---|---|---|---|---|---|
| Web Project | 91 | 202 | 4 | 60 | 1211 | 199 |
| Test Project | 90 | 2 | 1 | 5 | 16 | 3 |

Table: Code Coverage Metrics from Visual Studio Code Metrics Results.

## 6 Deployment metrics

Build Pipeline Jobs:

- Gent job1: Initial configuration of Azure pool, Ubuntu-16.04 Image
- Check out source code from TFCVC
- Acquires a specific version of the .NET Core SDK
- Build, test, package, or publish web project

The following matrix table shows how long was deployment time for Build Pipeline and Release Pipeline.

| | Build Pipeline | Release Pipeline |
|---|---|---|
| Deployment Time | 2m 10s | 13s |

Table: Deployment Metrics for Build and Release Pipeline.

# 7 Continuous Integration(CI)

## 7.1 CI setup

As mentioned, this website is created in Visual Studio (VS) utilising Asp.Net MVC Core Application. One of the most interesting features of VS is the Team Explorer and Source Control. Visual Studio Team Explorer has helped us add and integrate code changes to the main branch. We set up CI in VS through using Team Explorer and connected it to our main project. This allowed for the DevOps project to connect with the logged in user. Once this was achieved, we used “Source Control Explorer” from the View/Other Windows and mapped the local folder with the DevOps project. After this we copied and pasted the VS project to the folder and added the files to our Source Control and synchronised them together. It is then possible to view these changes in a new window where all the changes or new code can be check in to the Master or Branch. It is also possible to add the Task ID with every Check-In by using ”Add Work Item by ID”. This Task ID can be found in Azure DevOps Boards/Work Items. Moreover, in the” Comment” option, it is possible to add more details of the added task. Once the changed code is pushed to the Master project in Azure DevOps, a successful message can be shown in the Team Explorer. All the sources and the changesets can be found in the Azure DevOps under Repositories.

## 7.2 Detailed Presentation of the Azure Build pipeline

An automated build is achieved and hosted through Azure Web Services within our project and can be viewed in our Azure DevOps board. To create a build pipeline, we used “Pipeline” within Azure DevOps. As our project uses Team Foundation Version Control, TFVC is selected. After that ASP.NET Core template is selected to build and test web applications. This opens a new window listed with Restore, Build, Test, Publish, Publish Artifact option for the build pipeline. At first, use .Net Core ADK is added by clicking the plus icon in the Agent Job. .Net core opens, it is needed to add the version. In this case, 2.1.302 is written. Change default build pipeline name to ”¡project name ¿ Build”. This can help to find and distinguish between the different pipelines. Next, we enabled continuous integration under the Triggers option so that any changes in the pipeline can trigger the build pipeline. We saved all configurations and run the pipeline. A summary is shown during a pipeline running within the Azure DevOps board. More details of each job can be seen by clicking the Agent Job1 icon in the queued jobs. It can show the status of jobs like Build, Test, Publish,

Publish Artifact. It also shows the test results. As in this project, a test project is added, the test is passed successfully. Finally, if there is no error occurring, the Build pipeline is successfully created, and a success status is shown.

### 7.3 Azure Release Pipeline

We created a Release Pipeline utilising “Releases” in Azure DevOps. This opens a new window where Stage and artifacts are configured. We added Azure App Service deployment from the feature list. Stage requires to add where Stage name can be changed. This stage is used later when the app service plans would be defined. After that, an artifact is added with the source of the build pipeline which is created in the previous stage. Continuous integration is enabled here by pressing the thunder button in the artifact section. After enabling CI, a correct check mark would be visible. In the tasks option, along with stage name, a valid azure subscription, app type, and app service plan are configured. Note that the App service plan and Azure subscription must be created prior to create a pipeline. Artifacts source can be configured under the File Transform Variable Substitution Option. This would allow selecting a .Zip file. Now, a deployment method (Web Deploy) is selected with some other additional features like take the app offline, remove additional files at destination, rename locked files. Under Options, release name format can be defined for example adding ”Build.BuildNumber” in front of the release version can be shown the current build number with the release version. After that, save all of the configurations with appropriate comments like ”Initial Release created”. Pressing OK would show the release pipeline Artifact and Sages that have been configured so far. Now, click to create a new release and this would show the pipeline and artifact that have been created. Once the create button is pressed, a release would be created. Finally, it shows the progress dashboard of the release pipeline and successful notification. The overall project pipelines can be observers from the DevOps Overview pages too.

## 8 Explanation of the Continuous Delivery (CD) setup

Firstly, we created an organization in Azure DevOps. Under the organization, a project is created. Once the project is created, it shows the DevOps portal, where it is possible to add project members, add project information in the wiki, add members’ tasks, lookup or manage project repositories, add or delete pipelines as well as test plans and artifacts. This portal makes it possible to control the access of the members, contributors, administrations. It is also possible to add and delete a policy. In Azure Boards and

Work Items, each of the project tasks is created. It is possible to assign each task to the member, with the options such as the state of the task and the priority. When all tasks are created, it can be shown in the list of all tasks. Project features are added and linked with other tasks that are related to each feature. In the backlog opting, it can be observed on a hierarchical view of each feature and connected tasks. On the Boards, each of the project feature statuses can be observed as New or In Progress or Done. This can help to keep track of the program of each work item more efficiently.

# 9 Functionality/Features of the web app

## 9.1 COVID-19 Latest Update

In this website, COVID-19 update regarding confirmed, recovered, and death cases have been shown by using COVID-19 API. Global summary of COVID-19 can be shown in a nice front-end from “https://news.google.com/covid19/map?hl=en-USgl=USceid=US:en”.JSON formatted data can be received from this API ”https://corona.lmao.ninja/v2/all”. It is also possible to find all the confirmed cases for a country like Sweden.

## 9.2 Contribution to Novelty Helping (Unique novel)

The uniqueness of this web app regarding novelty is that it has connected both small Facebook group to collect support for the COVID19 positive patients as well as promoting WHO Donation fund for the covid19 affected people. Moreover, others helping and supporting the research work or employees have been linked up on the web page.

## 9.3 Attract Support from other KTH students

Student Portal: In this section, KTH students can be able to get information regarding information and access in the canvas too. Because of the lockdown situation, Zoom has been linked up for the KTH students. For future students, information regarding courses and contact information has been provided in under each program.

## 9.4 Attract Support from Society: Community Portal

People from different communities can get help from the Community portal according to their municipality. There is an option for each section. In the first section, people can get their Emergency help by visiting the linked address. For the COVID-19 test kit or any medical assistance, ”COVID-19? Support” can be referred to. If someone can learn more in the ”COVID-19:

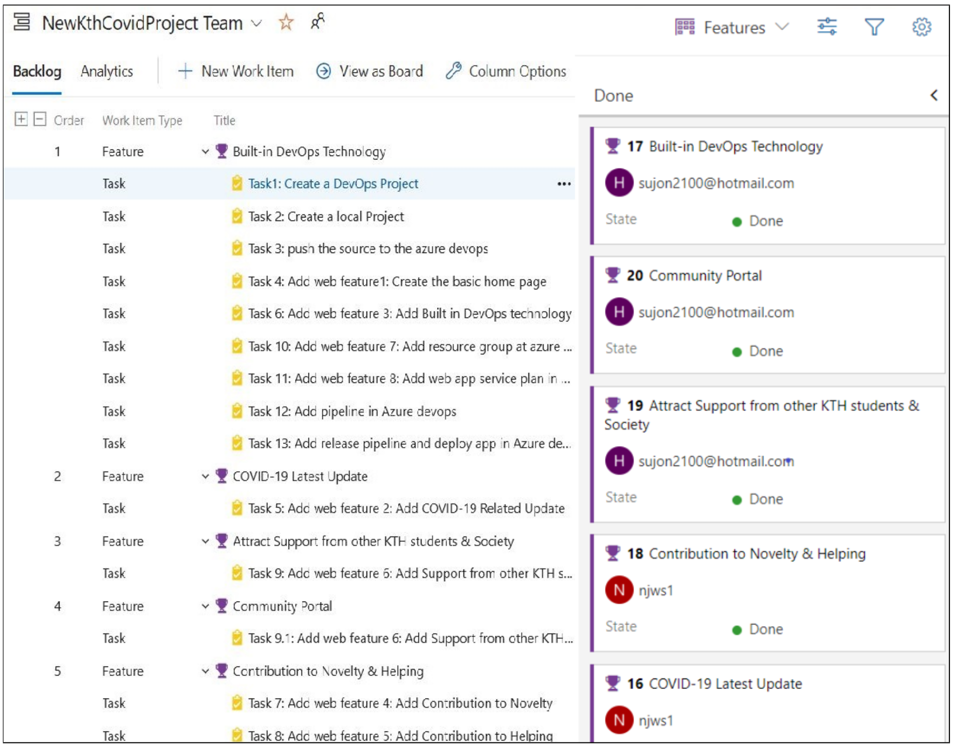
Learn more" section about the symptoms of this disease and all related issues. "COVID-19 Update" section can give all current information regarding different cities and communities.

## 10 Dissemination plans: where did/will we advertise the website?

We can advertise this website on social media like Facebook, Instagram, Linkedin. This would help to let people know more about the Help Fight COVID19 project and encourage people to attract to use and contribute to society. We can advertise on our COVID19 Facebook page(https://m.facebook .com/groups/815512272277849).

## 11 Contribution

- This COVID-19 project can provide Corona-virus related updates such as confirmed, recovered, and death.
- Implementation of the DevOps technology such as Azure DevOps for CI/CD.
- Promoting fundraising for the COVID19 patients throughout WHO external web service, as well as a concept regarding integration with the local Facebook groups to collect support from everyone, could help COVID19 affected people.
- This website could attract KTH students as they can use for there current studies as well as all the latest corona update from KTH administration. They can also join the Facebook group to support COVID19 affected people.
- To manage the development of the web app, we created an Azure task board, highlighting the technical contributions that needed to be implemented, this is shown below.



| Order | Work Item Type | Title |
|---|---|---|
| 1 | Feature | Built-in DevOps Technology |
| | Task | Task1: Create a DevOps Project |
| | Task | Task 2: Create a local Project |
| | Task | Task 3: push the source to the azure devops |
| | Task | Task 4: Add web feature1: Create the basic home page |
| | Task | Task 6: Add web feature 3: Add Built in DevOps technology |
| | Task | Task 10: Add web feature 7: Add resource group at azure ... |
| | Task | Task 11: Add web feature 8: Add web app service plan in ... |
| | Task | Task 12: Add pipeline in Azure devops |
| | Task | Task 13: Add release pipeline and deploy app in Azure de... |
| 2 | Feature | COVID-19 Latest Update |
| | Task | Task 5: Add web feature 2: Add COVID-19 Related Update |
| 3 | Feature | Attract Support from other KTH students & Society |
| | Task | Task 9: Add web feature 6: Add Support from other KTH s... |
| 4 | Feature | Community Portal |
| | Task | Task 9.1: Add web feature 6: Add Support from other KTH... |
| 5 | Feature | Contribution to Novelty & Helping |
| | Task | Task 7: Add web feature 4: Add Contribution to Novelty |
| | Task | Task 8: Add web feature 5: Add Contribution to Helping |

**17** Built-in DevOps Technology
H sujon2100@hotmail.com
State ● Done

**20** Community Portal
H sujon2100@hotmail.com
State ● Done

**19** Attract Support from other KTH students & Society
H sujon2100@hotmail.com
State ● Done

**18** Contribution to Novelty & Helping
N njws1
State ● Done

**16** COVID-19 Latest Update
N njws1

Figure: Project Contribution.

## 12 References

- https://dev.azure.com/KthProject2020/NewKthCovidProject
- https://news.google.com/covid19/map?hl=en-USgl=USceid=US:en
- https://corona.lmao.ninja/v2/all
- https://api.covid19api.com/country/sweden/status/confirmed
- https://portal.azure.com/home
- https://visualstudio.microsoft.com/
- https://covid19responsefund.org/
- https://m.facebook.com/groups/815512272277849
- https://www.canada.ca/en/public-health/services/publications/diseases-conditions/covid-19-information-essential-service-workers.html
- https://www.canada.ca/en/public-health/services/publications/diseases-conditions/help-reduce-spread-covid-19.html

- https://kth.instructure.com/courses/
- https://www.kth.se/en/om/kontakt
- https://polisen.se/en/the-swedish-police/contacting-the-police/
- https://www.vardcentraler.sll.se/
- https://www.folkhalsomyndigheten.se/the-public-health-agency-of-sweden/communicable-disease-control/covid-19/about-the-disease/
- https://intra.kth.se/en/campus/sakerhet/kris/corona/information-till-anstallda-med-anledning-av-coronaviruset-1.965906
- https://intra.kth.se/en/campus/sakerhet/kris/corona/information-till-studenter-med-anledning-av-coronaviruset-1.965905
- https://www.youtube.com/embed/k8vuUJ7hVLY
- https://www.youtube.com/embed/JCw-B4Ig3DQ