Document

เรื่อง โปรแกรมแนะนำการแก้โจทย์ปัญหากลศาสตร์การเคลื่อนที่

รายวิชา 204362

จัดทำโดย...

นายอนันต์ อุ่นทา

รหัสนักศึกษา 560510679

เสนอ...

อาจารย์ ดร.อารีรัตน์ ตรงรัศมีทอง

ภาคเรียนที่ 2

ปีการศึกษา 2558

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเชียงใหม่

สารบัญ

**ชื่อเรื่อง**

Proposal

Business Flow Diagram

Use Case Diagram

Use Case Specification

UI

Sequence Diagram

State Machine Diagram

Component Diagram

**หน้า**

1

4

5

6

9

10

12

13

**แบบเสนอ Project Proposal 204362 ภาคการศึกษาที่ 2 ปี การศึกษา 2558**

**1. รหัส** 560510679  **ชื่อ – สกุล**  นายอนันต์ อุ่นทา

**อาจารย์ที่ปรึกษาการค้นคว้าอิสระ** อาจารย์ ดร.ศุภกิจ อาวิพันธุ์

**2. ชื่อเรื่องภาษาไทย** โปรแกรมแนะนำการแก้โจทย์ปัญหากลศาสตร์การเคลื่อนที่

**ชื่อเรื่องภาษาอังกฤษ** Kinematics Problems Solving Guide Program

**3. หลักการและเหตุผล**

วิชาฟิสิกส์เป็นวิชาหนึ่งที่มีความยากในการทำความเข้าใจในวิชานี้ เพราะวิชาฟิสิกส์ต้องใช้การจำลองเหตุการณ์อย่างมากในการนึกถึงสิ่งต่างๆ ที่เกิดขึ้น เพื่อใช้ช่วยในการคำนวณ หากเราไม่สามารถจำลองเหตุการณ์ที่เกิดขึ้น จากสิ่งที่โจทย์ปัญหากำหนดให้ เราก็ไม่สามารถที่จะแก้ปัญหาในโจทย์ข้อนั้นได้เลย เพราะว่าในการแก้โจทย์ปัญหาฟิสิกส์นั้นต้องใช้การจำลองเหตุการณ์ว่าหากเกิดเหตุการณ์แบบนี้ควรใช้สูตรฟิสิกส์ข้อไหนในการแก้ปัญหา อีกทั้งหากเราสามารถจำลองเหตุการณ์ที่เกิดจากสิ่งที่โจทย์ปัญหากำหนดได้ ในส่วนของการคำนวณก็ยังคงเป็นปัญหา เพราะมีการแก้สมการที่มีความซับซ้อนในระดับหนึ่งซึ้งอาจทำให้เกิดความผิดพลาดระหว่างการคำนวณได้ ทำให้ในการแก้ปัญหาของโจทย์ปัญหาวิชาฟิสิกส์มีโอกาสสูงที่จะเกิดความผิดพลาดในการหาคำตอบ เพราะฉะนั้นแล้วผู้จัดทำจึงมีแนวคิดในการพัฒนาโปรแกรมแนะนำการแก้โจทย์ปัญหากลศาสตร์การเคลื่อนที่ เพื่อช่วยให้นักเรียน นักศึกษา ได้ใช้โปรแกรมนี้ในการตรวจสอบวิธีการในการแก้ปัญหา ตรวจสอบความถูกต้องของคำตอบที่ตนเองทำการแก้โจทย์ปัญหา และเพิ่มทักษะในการแก้ปัญหาของโจทย์ปัญหาวิชาฟิสิกส์ เพราะโปรแกรมนี้สามารถแสดงผลลัพธ์ของโจทย์ปัญหา และสามารถแสดงขั้นตอนวิธีในการแก้ปัญหาของโจทย์นั้นๆ อีกทั้ง บุคคลทั่วไปที่ต้องการจะศึกษาในวิชาฟิสิกส์ก็ยังสามารถใช้โปรแกรมนี้เพื่อเป็นสื่อการเรียนการสอนได้อีกด้วย รวมทั้งอาจารย์ก็สามารถนำโปรแกรมนี้ไปช่วยในการตรวจคำตอบของข้อสอบต่างๆ ช่วยลดความผิดพลาดที่เกิดขึ้นได้

**4. วัตถุประสงค์ของโครงงาน**

4.1 ช่วยแสดงคำตอบของโจทย์ปัญหากลศาสตร์การเคลื่อนที่

4.2 ช่วยเพิ่มความเข้าใจในการแก้โจทย์ปัญหากลศาสตร์การเคลื่อนที่ โดยการแสดงวิธีการแก้ปัญหา

4.3 ช่วยในการแนะนำวิธีการแก้โจทย์ปัญหากลศาสตร์การเคลื่อนที่

**5. ประโยชน์ที่จะได้รับจากการศึกษาเชิงประยุกต์**

ประโยชน์ที่จะได้รับในโครงงานนี้คือ นักเรียน นักศึกษา บุคคลทั่วไป จะสามารถใช้ประโยชน์จากโปรแกรมที่พัฒนาขึ้นไปช่วยในการคำนวณหาคำตอบของโจทย์ปัญหาเพื่อเสริมความเข้าใจ รวมทั้งได้ทราบถึงวิธีการแก้ปัญหาของโจทย์ดังกล่าวจากการที่โปรแกรมแสดงวิธีทำการหาคำตอบด้วยเพื่อฝึกฝนทักษะในการแก้ปัญหาโจทย์ปัญหานั้นๆ หรืออาจจะนำไปใช้ในการตรวจสอบคำตอบที่ถูกต้องในการฝึกทำโจทย์ปัญหาได้ อีกทั้งยังสามารถช่วยให้ ครูอาจารย์ สามารถตรวจสอบคำตอบของโจทย์ปัญหาได้ เพราะการที่การแก้ปัญหาโดยมนุษย์อาจเกิดข้อผิดพลาดได้ ดังนั้นเราสามารถใช้โปรแกรมในการช่วยตรวจสอบคำตอบเพื่อผลลัพธ์ที่ถูกต้องได้

**6. ขอบเขตของโครงงาน**

**6.1 ขอบเขตทางสถาปัตยกรรม**

6.1.1 เป็นโปรแกรมออฟไลน์ (Stand Alone)

6.1.2 ฮาร์ดแวร์ (Hardware) ที่ใช้ในการพัฒนาระบบ ประกอบด้วย

- คอมพิวเตอร์แล็ปท็อป (Laptop Computer)

- หน่วยประมวลผล อินเทลคอร์ ไอ i7-4712 เอ็มคิว 2.30 จิกะเฮิรตซ์ (Intel® core™ i7-4712MQ @2.30GHz)

- หน่วยความจำหลัก (RAM) 8 จิกะไบต์(8 GB)

- หน่วยความจำสำรอง (HDD) 1 เทระไบต์ (1 TB)

6.1.3 ซอฟต์แวร์ (Software) ที่ใช้ในการพัฒนาระบบ ประกอบด้วย

- ระบบปฏิบัติการ ลีนุกซ์ มิ้นท์ 17.3 เอกซ์เอฟซีอี 64 บิต (Linux Mint 17.3 xfce 64 bit) เป็นระบบปฏิบัติการที่ไม่เสียค่าใช้จ่ายในการใช้งานมีความรวดเร็ว สะดวกต่อการใช้งานในการพัฒนาโปรแกรม

- ไอดีแอลอี ไพทอน 3.5 32 บิต (IDLE python 3.5 32bit) เป็นเครื่องมือในการเขียนโปรแกรม

6.1.4 ภาษาที่ใช้ในการพัฒนาระบบ

- ไพทอน (Python) เป็นภาษาที่มีฟังก์ชันให้เลือกใช้มากมายทำให้มีความสะดวกมากในการพัฒนาโปรแกรม มีความรวดเร็ว และเป็นภาษาที่มีลักษณะการเขียนที่เข้าใจง่ายไม่ยุ่งยาก

**6.2 ขอบเขตของระบบงาน**

6.2.1 ผู้ใช้ทั่วไป

- ผู้ใช้ต้องนำเข้าข้อมูล เป็นตัวแปร และค่าของตัวแปร รวมทั้งตัวแปรที่โจทย์ต้องการให้หาคำตอบของโจทย์ปัญหาวิชาฟิสิกส์ มี โดยมีหน่วยของค่าของตัวแปรที่โจทย์ต้องการคำตอบ และ ค่าของตัวแปรที่โจทย์กำหนดให้กำหนดไว้ชัดเจน มีส่วนเนื้อหาของโจทย์ปัญหาเป็นเรื่องของการเคลื่อนที่เท่านั้น แล้วโปรแกรมจะแสดงวิธีการคำนวณ เพื่อหาคำตอบของโจทย์ปัญหานั้น

6.2.2 ขอบเขตของการประมวลผล

- การคำนวณของโจทย์ปัญหา เมื่อรับได้ตัวแปรที่โจทย์ต้องการคำตอบ ตัวแปรที่โจทย์กำหนดให้ รวมทั้งค่าของตัวแปร จากผู้ใช้ทั่วไป จากนั้น จะทำการคำนวณว่า ควรใช้สูตรใดในการหาคำตอบ ทำการแก้สมการตามสูตรนั้นๆ และแสดงผลของคำตอบพร้อมวิธีการแก้สมการของสูตรแต่ละข้อ

**6.3 ขอบเขตของข้อมูล**

6.3.1 ข้อมูลของตัวแปรของสมการที่ใช้ในการแก้ปัญหา

6.3.2 ข้อมูลของสูตรที่ใช้ในการคำนวณจำนวนไม่เกิน 20 สูตร

Business Flow

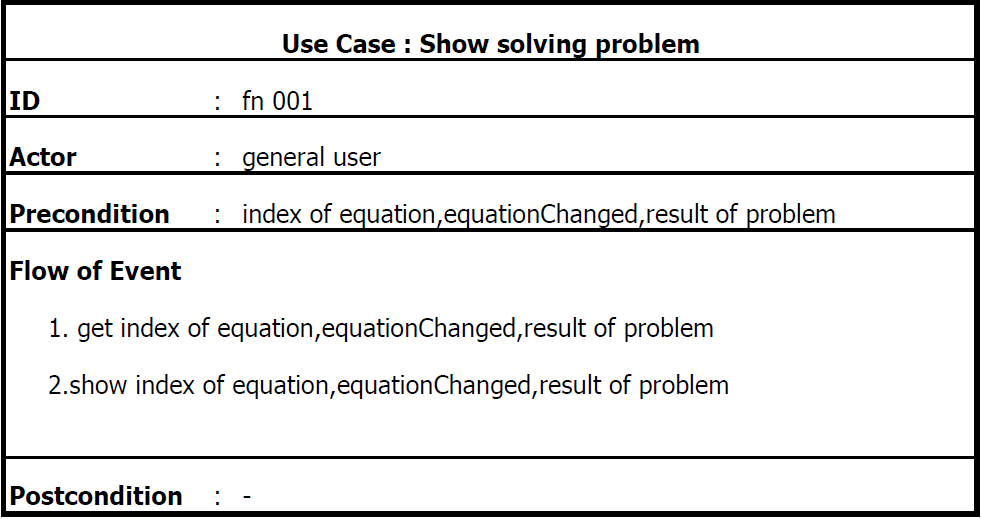
Use Case



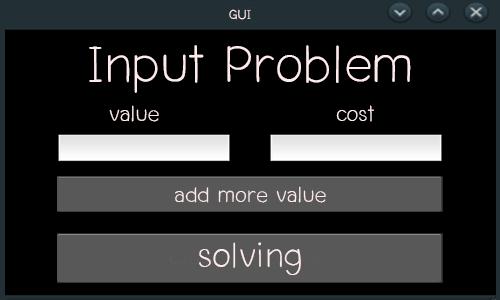
Class specification

|  |  |  |
| --- | --- | --- |
| **Use Case : Input problem** | | |
| **ID** | : | st001 |
| **Actor** | : | general user |
| **Precondition** | : | - |
| **Flow of Event** |  |  |
| 1.get problem |  |  |
| 2.check syntax problem | | |
| 3.return problem | | |
| **Postcondition** | : | value,valueFind,cost |
|  | | |
| **Use Case : Find Equation** | | |
| **ID** | : | fe001 |
| **Actor** | : | program |
| **Precondition** | : | value,valueFind,valueOfEquation[] |
| **Flow of Event** |  |  |
| 1.get value,valueFind,valueOfEquation[] | | |
| 2 call function callEquation(value,valueFind,valueOfEquation): | | |
| list a=value+valueFind | | |
| if a in valueOfEquation[]: | | |
| return valueOfEquation[valueOfEquation.index(a)] | | |
| else : |  |  |
| a=min( valueOfEquation - (value+valueFind) ) | | |
| call function calEquation(value,valueFind,valueOfEquation): | | |
| 3 return list index of equation | | |
| **Postcondition** | : | index of equation |

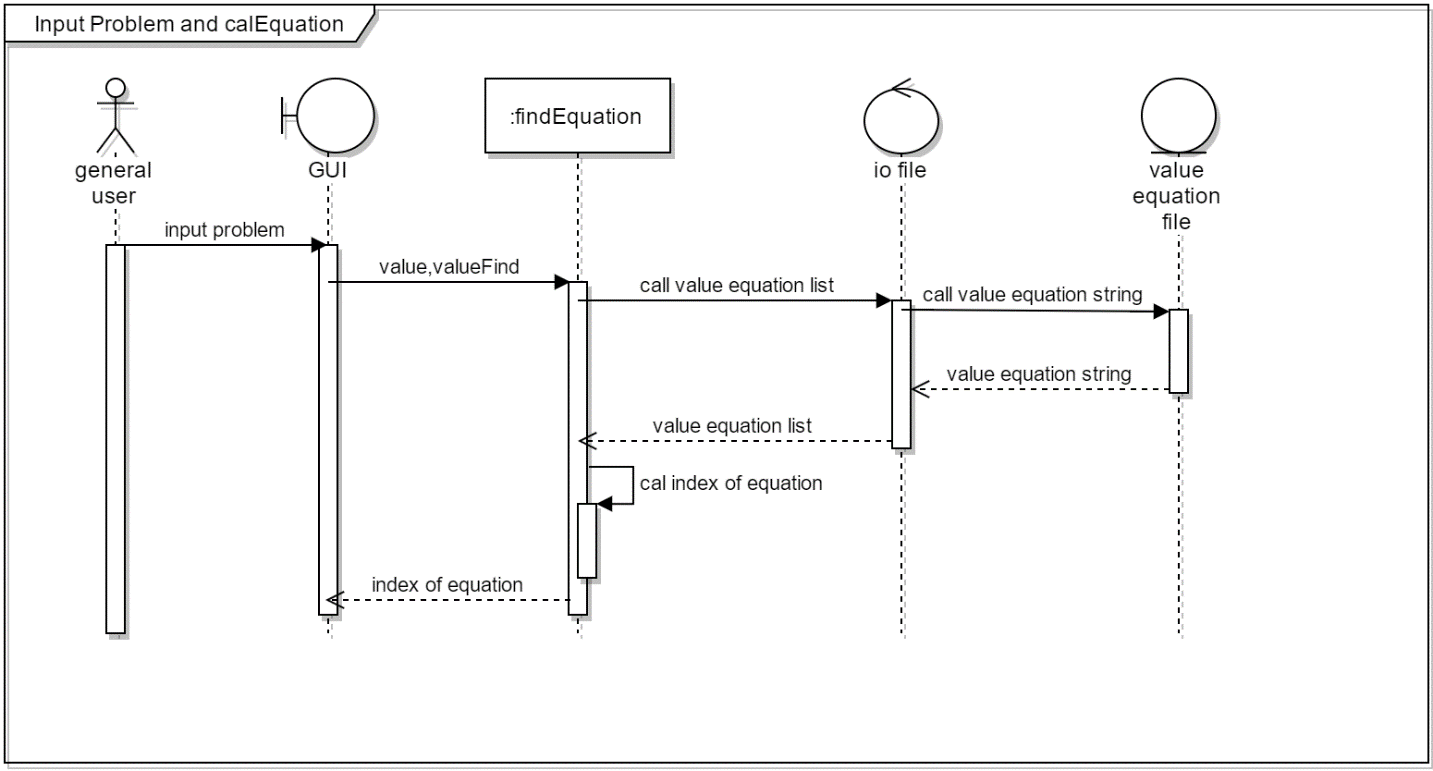
|  |  |  |
| --- | --- | --- |
| **Use Case : Find Equation** | | |
| **ID** | : | fe002 |
| **Actor** | : | program |
| **Precondition** | : | value,valueFind,index of equation,cost,equation |
| **Flow of Event** |  |  |
| 1 get value,valueFind,equation,index | | |
| 2 call function changEquation(value,valueFind,equation[index]): | | |
| left=equation[:equation.index('=')-1] | | |
| right=equation[equation.index('=')+1:] | | |
| if valuefind in right | | |
| swap left,right | | |
| if value in left >1 | | |
| return call function poly | | |
| return left+rigth+'=0' | | |
| give value of left is not valueFind swap to rigth | | |
| return left+'='+right | | |
| 3 get result of changEquation function to equationChanged value | | |
| 4 call function calValue(equationChanged,value,cost) | | |
| subE=gorup(equation) | | |
| if subE=1:return cost[cost.index(subEquation[0]) | | |
| if subE=2:return operation.index(subE[0])calValue(subE[1]) | | |
| else:return calValue(subE[0]) operation.index(subE[1]) calValue(subE[2]) | | |
| 5 get result of calValue to result of problem | | |
| **Postcondition** | : | equationChanged,result of problem |

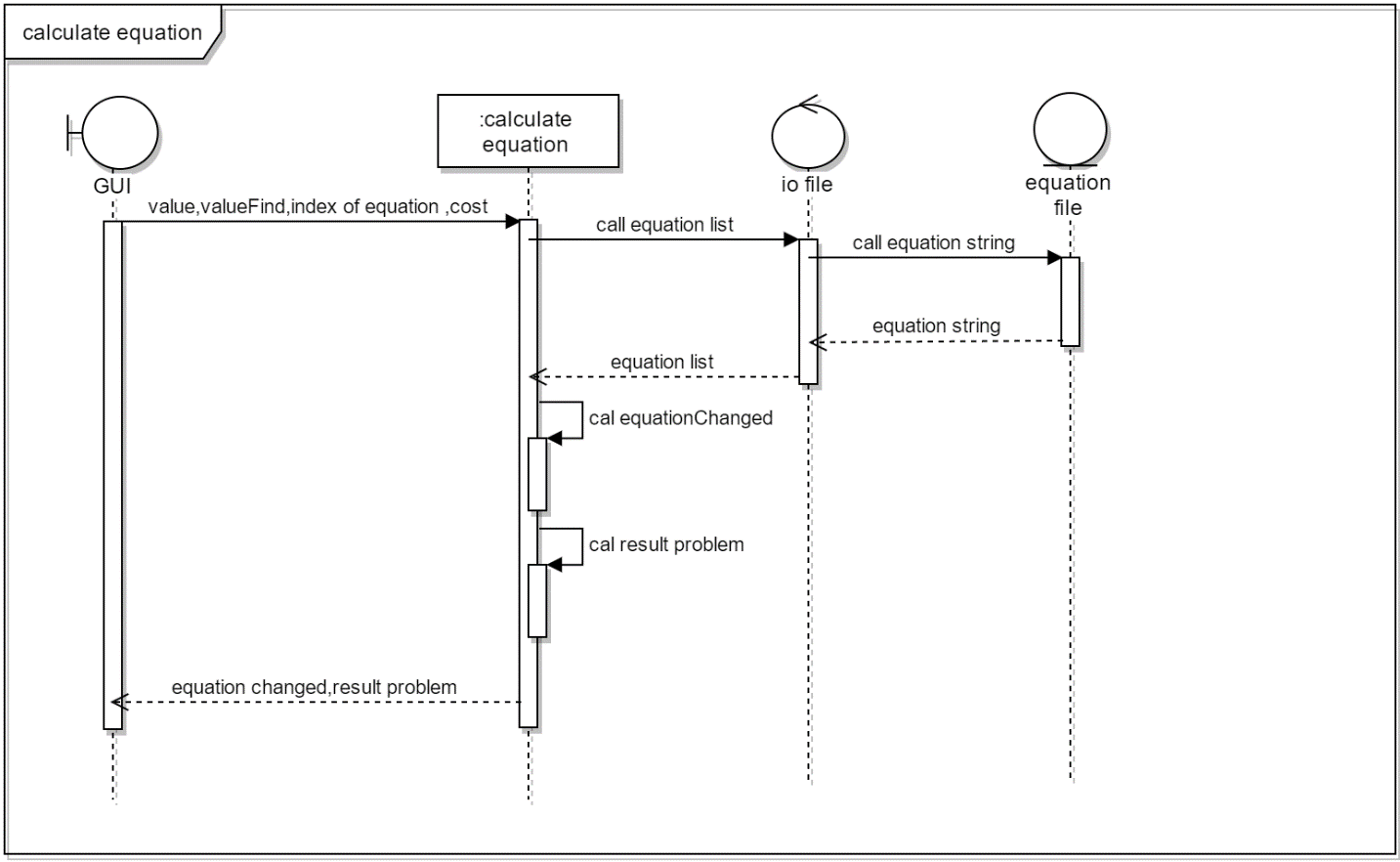


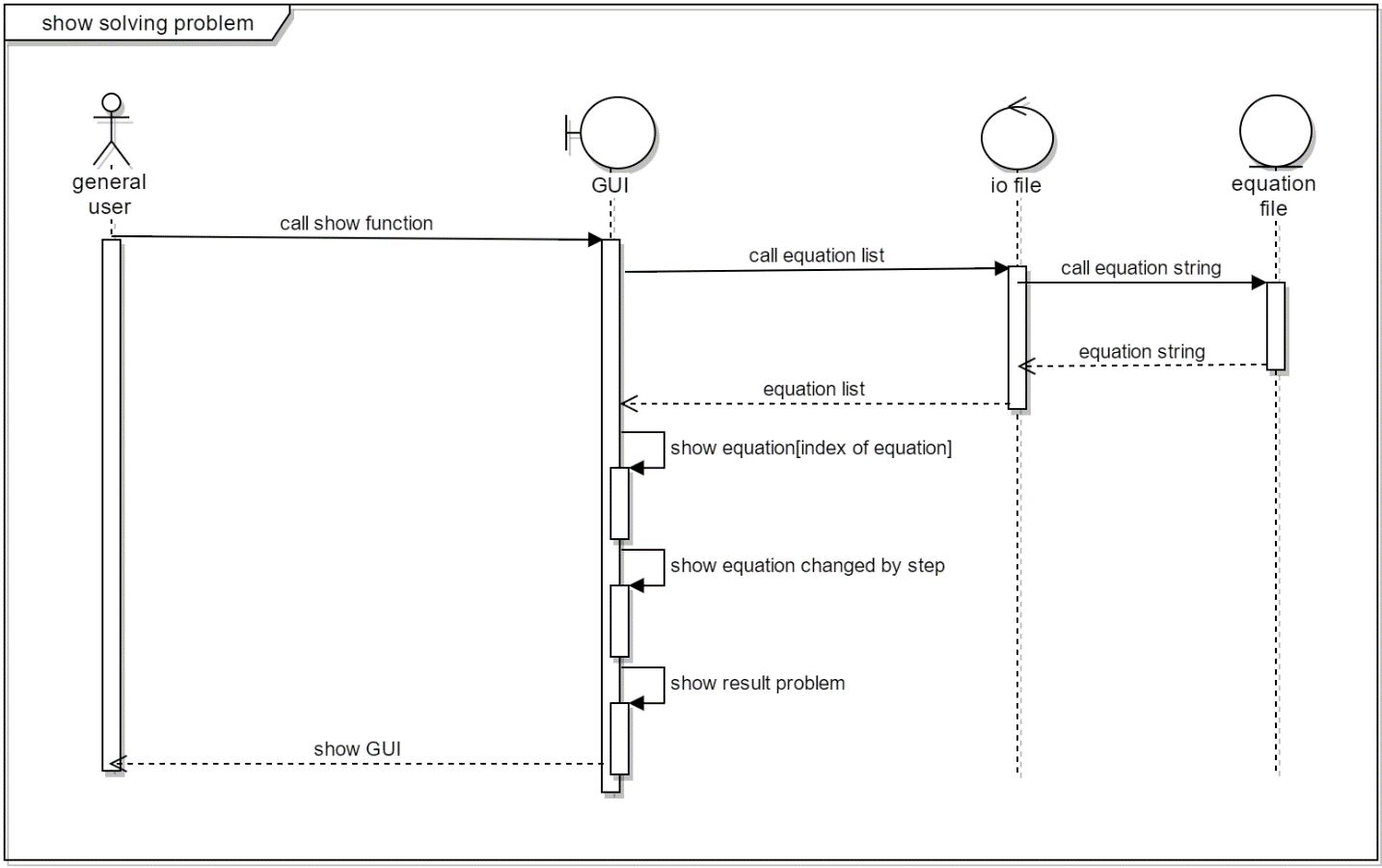
UI



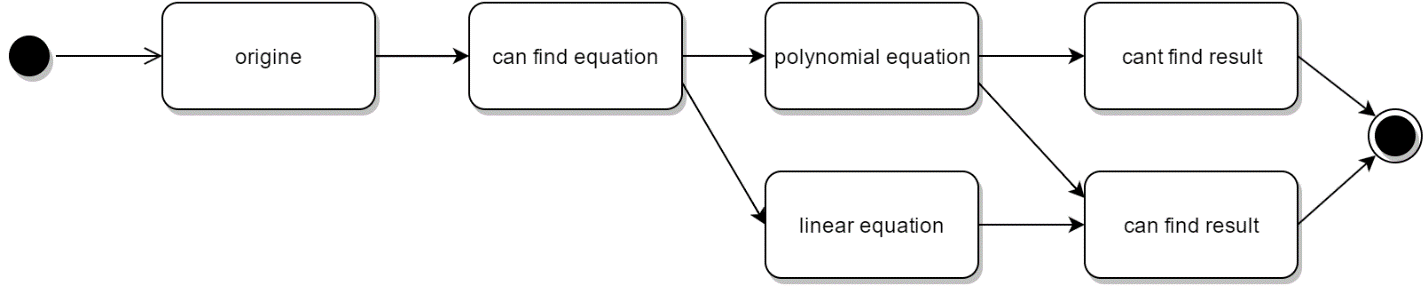
Sequence Diagram







State Machine



State machine of problem

Component Diagram

